

Implementación de un sistema de gestión de base de datos aplicado a los desarrollos de generación fotovoltaica y cultivo de las terrazas verdes del Edificio de Ingeniería Eléctrica

María Claudia Villarreal Ardila

Trabajo de Grado para Optar el título de Ingeniera Electrónica

Director

German Alfonso Osma Pinto

Doctor en Ingeniería Eléctrica

Codirector

María Alejandra Camargo Vila

Ingeniera Electrónica

Universidad Industrial de Santander

Facultad de Ingenierías Físicomecánicas

Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Bucaramanga

2019

Dedicatoria

A mi abuelita

Agradecimientos

Al profesor German Osma, por su calidad humana y profesional.

A los ingenieros María Alejandra Camargo y Darío Alejandro Riaño, por su instrucción, esfuerzo y dedicación.

A los miembros del grupo de investigación GISEL, por su colaboración y compañía.

A mi familia, por el amor incondicional y la felicidad que me brindan, especialmente a mis padres, por su ejemplo, paciencia y entrega.

A Dios, por mi familia.

Tabla de Contenido

	Pág.
Introducción.....	13
1. Objetivos.....	16
1.1 Objetivo General.....	16
1.2 Objetivos Específicos.....	16
2. Consideraciones generales.....	17
2.1. Sistema GRIPV.....	17
2.4. Herramientas de programación.....	21
3. Desarrollo del sistema de adquisición de datos.....	24
3.1. Comunicación a partir del protocolo modbus.....	24
3.2. Rutina para adquisición de datos.....	29
3.3. Pruebas de funcionamiento de la rutina de adquisición de datos.....	36
4. Diseño de la base de datos.....	44
4.1. Requerimientos.....	44
4.2. Modelo conceptual.....	44
4.3. Modelo relacional.....	48
4.4. Esquema.....	49

5.	Instalación del sistema e implementación del SGBD.....	53
5.1.	Herramientas de MySql	53
5.2.	Condiciones iniciales	54
5.3.	Instalación del sistema de adquisición de datos	57
5.4.	Pruebas del sistema implementado	61
5.5.	Mejoramiento del sistema y versión final.....	63
5.6.	Equipos y costos	66
6.	Conclusiones.....	68
7.	Recomendaciones	70
	Referencias Bibliográficas.....	71

Lista de Tablas

Tabla 1. Estructura de la trama de datos para consultas en el protocolo Modbus.....	25
Tabla 2. Códigos de función más utilizados en el protocolo Modbus.....	26
Tabla 3. Ejemplo de consulta mediante protocolo Modbus.	27
Tabla 4. Estructura de la trama de datos de las respuestas en el protocolo Modbus.	27
Tabla 5. Ejemplo de respuesta mediante protocolo Modbus.....	29
Tabla 6. Resultados de la comparación de datos extraídos por el AcuRev.	38
Tabla 7. Resultados de la comparación de datos extraídos por el Acuvim.	41
Tabla 8. Características de los ordenadores de placa reducida que cumplen con las especificaciones.	58
Tabla 9. Equipos y costos.	67

Lista de Figuras

<i>Figura 1:</i> Sistema GRIPV, terraza superior del Edificio de Ingeniería Eléctrica	17
<i>Figura 2:</i> Ubicación inicial de los elementos del sistema de medición en el Edificio de Ingeniería Eléctrica.	18
<i>Figura 3:</i> Acuvim II. (Corp, 2014)	19
<i>Figura 4:</i> AcuRev 2020. (Power, Meter, & Certified, 2000)	20
<i>Figura 5:</i> Red de comunicación Modbus	24
<i>Figura 6:</i> Ejemplo de comunicación usando el software Hercules SETUP.	28
<i>Figura 7:</i> Vista del documento de entrada del Acuvim	31
<i>Figura 8:</i> Interfaz gráfica de usuario.	32
<i>Figura 9:</i> Entradas y salidas de la rutina	33
<i>Figura 10:</i> Vista del documento de salida de la rutina	34
<i>Figura 11:</i> Visualización de algunos elementos de la rutina en el diagrama.	35
<i>Figura 12:</i> Gráficas de datos Modbus vs datos descargados para dos parámetros del Acuvim ..	38
<i>Figura 13:</i> Gráfica de la frecuencia medida por los dos equipos	43
<i>Figura 14:</i> Elementos del modelo conceptual	46
<i>Figura 15:</i> Diagrama entidad-relación de la base de datos	47
<i>Figura 16:</i> Conceptos aplicados a la tabla Equipos de la base de datos	48
<i>Figura 17:</i> Zona de trabajo en MySQL Workbench	50
<i>Figura 18:</i> Esquema de la base de datos realizado en MySQL Workbench	51
<i>Figura 19:</i> Script SQL generado a partir de la opción "Forward Engineer" de MySQL Workbench.	52
<i>Figura 20:</i> Medidores instalados	56

<i>Figura 21:</i> Adaptador RS485 a USB	57
<i>Figura 22:</i> Tarjetas adquiridas para la implementación	58
<i>Figura 23:</i> Hardware del sistema de adquisición de datos.....	¡Error! Marcador no definido.
<i>Figura 24:</i> Interfaz de usuario de la rutina para adquisición de datos	60
<i>Figura 25:</i> Mediciones almacenadas en la base de datos sin desfase de tiempo.....	62
<i>Figura 26:</i> Prueba de funcionamiento de dos rutinas corriendo simultáneamente en la misma tarjeta.....	65
<i>Figura 27:</i> Disposición final de los elementos del sistema de adquisición de datos.....	66

Resumen

Título: Implementación de un sistema de gestión de base de datos aplicado a los desarrollos de generación fotovoltaica y cultivo de las terrazas verdes del Edificio de Ingeniería Eléctrica¹

Autor María Claudia Villarreal Ardila²

Palabras clave Bases de datos, SGBD (sistemas de gestión de base de datos), sistemas GRIPV (*Green Roof Integrated Photovoltaics*), protocolo Modbus.

Descripción: El sistema de generación fotovoltaica integrado con vegetación, instalado en la terraza superior del Edificio de Ingeniería Eléctrica; además de ser una alternativa de generación energética, ha sido utilizado con fines académicos y en torno a él se desarrollan proyectos de investigación que buscan evaluar el desempeño de este tipo de sistemas en la zona tropical, dado que es una tecnología verde emergente en el mercado de las edificaciones sostenibles.

Se identificó la necesidad de generar una base de datos (BD) que se actualice de forma permanente, esto implica el desarrollo de un mecanismo para la adquisición automática de la información producto de la monitorización del sistema GRIPV, y la implementación de un sistema de gestión de base de datos (SGBD).

Se desarrolló una rutina en JAVA que solicita información a los equipos mediante protocolo modbus en intervalos de tiempo seleccionados por el usuario, esto facilita la sincronización y la centralización de los datos, y resuelve inconvenientes asociados a las dinámicas de operación de los equipos. Se diseñó la base de datos y finalmente se implementó el SGBD. Los valores de variables eléctricas almacenados son entregados por seis medidores digitales de dos clases diferentes; Acuvim II y Acurrev2020. Como dispositivos maestros se usaron tres tarjetas Raspberry Pi 3.

Se desarrollaron pruebas para identificar errores de funcionamiento, una vez estos fueron corregidos, se obtuvo un sistema que adquiere datos precisos de forma sincronizada, y los almacena en una base de datos, lo cual facilita el acceso a la información, aumenta la cantidad de variables eléctricas que pueden monitorizarse; puesto que es independiente de la memoria interna de los equipos, disminuye ampliamente la cantidad de horas-hombre necesarias para la descarga de los datos y aumenta la seguridad de la información.

¹ Trabajo de grado

² Facultad de Ingenierías Fisicomecánicas, Escuela de ingeniería Eléctrica, Electrónica y de Telecomunicaciones. Director: Dr. German Alfonso Osma Pinto, Codirector: Ing. María Alejandra Camargo Vila.

Abstract

Title: Implementation of a database management system applied to the development of photovoltaic generation and cultivation of the green terraces of the Electrical Engineering Building³

Author María Claudia Villarreal Ardila⁴

Keywords Database, DBMS (Database management system), GRIPV (*Green Roof Integrated Photovoltaics*), modbus protocol.

Description: The Green Roof Integrated Photovoltaic installed on the upper terrace of the Electrical Engineering Building to be an alternative of energy generation as academic purposes was a research project developed to evaluate the performance of this type of systems in the tropical zone as an emerging green technology in the Bussines of sustainable buildings.

We identified the need of generating a database (DB) that can actualize it-self, which included the development of a mechanism for the automatic acquisition of information resulting from the monitoring of the GRIPV system and the implementation of a DataBase Management System (DBMS).

A JAVA routine was developed, the routine requests information from the equipment using a modbus protocol at intervals of time selected by the user, this facilitates the synchronization and centralization of the data, and resolves problems associated with the equipment operation dynamics. The database was designed and finally the DBMS was implemented. The values of stored electrical variables are delivered by six digital meters of two different classes; Acuvim II and Acurrev2020. Three Raspberry Pi 3 Single-board computers were used as master devices.

Tests were developed to identify operating errors, once they were corrected, a system was obtained that acquires accurate data in a synchronized manner, and stores them in a database, which facilitates access to information, increases the number of variables electrical that can be monitored; Since it is independent of the internal memory of the equipment, it greatly decreases the number of man-hours needed for the download of the data and increases the security of the information.

³ Bachelor Thesis

⁴ Facultad de Ingenierías Fisicomecánicas, Escuela de ingeniería Eléctrica, Electrónica y de Telecomunicaciones. Director: Dr. German Alfonso Osma Pinto, Codirector: Ing. Maria Alejandra Camargo Vila.

Introducción

En la actualidad, la terraza superior del Edificio de Ingeniería Eléctrica de la Universidad Industrial de Santander cuenta con varias aplicaciones verdes, entre las cuales se encuentra un sistema de generación fotovoltaica integrado con vegetación llamado GRIPV (Green Roof Integrated Photovoltaics); que combina paneles fotovoltaicos y techos verdes buscando aprovechar el efecto de sombreado de los primeros y el efecto de enfriamiento de los segundos.

El sistema fotovoltaico (FV) está conformado por: paneles FV, micro-inversores integrados a la red del edificio y un sistema de monitorización para variables eléctricas y ambientales. Además de ser una alternativa de generación energética, el sistema GRIPV ha sido utilizado con fines académicos, en el marco del proyecto COLCIENCIAS titulado “ Viabilidad técnica en la implementación de sistemas fotovoltaicos (FV) integrados con vegetación como estrategia de generación distribuida y horticultura en entornos urbanos de clima cálido tropical”, ejecución 2017-2019.

Entre los esfuerzos de investigación desarrollados en el marco de este proyecto COLCIENCIAS se encuentra el trabajo de investigación de Maestría en Ingeniería de Telecomunicaciones titulado “Plataforma de servicios basada en IoT para el monitoreo y control de un sistema de generación fotovoltaica (FV) integrado con vegetación”, que busca aprovechar el Internet de las Cosas para facilitar el análisis del desempeño histórico y en tiempo real del sistema GRIPV, permitiendo la visualización online y la descarga simultánea de datos correspondientes a variables eléctricas, térmicas y climáticas. Para el desarrollo de este trabajo de maestría se identificó la necesidad de generar una base de datos(BD) que se actualice de forma permanente.

Debido a la cantidad y a la diversidad de variables, es necesario emplear varios equipos de medición con dinámicas de operación propias; esto dificulta la adquisición y la integración de los datos, puesto que cada tipo de medidor opera con un software distinto para visualización y descarga de información.

Específicamente, este trabajo de grado busca apoyar la investigación mencionada, por lo que su objetivo es implementar un sistema de gestión de base de datos (SGBD) para la adquisición y almacenamiento automático de la información producto de la monitorización del sistema GRIPV, solucionando los inconvenientes asociados a las dinámicas de operación de los equipos.

Se tienen los siguientes objetivos específicos:

1. Desarrollar un sistema de adquisición de datos
2. Realizar el esquema e instancia de la base de datos
3. Implementar un SGBD para la creación y administración de la BD.

El sistema de adquisición de datos recolecta valores de variables eléctricas entregados por seis medidores de dos clases diferentes; Acuvim y AcuRev, sin embargo, se buscó que este sistema sea versátil, para facilitar la inclusión de otros dispositivos diversos y ante posibles mejoras.

La metodología establecida para la ejecución de este trabajo de grado se conformó por tres fases; inicialmente, se desarrolla el sistema de adquisición de datos automático; posteriormente, se diseña una base de datos que almacene la información extraída y, finalmente, se implementa un SGBD y se realiza la instalación y puesta en marcha de todo el sistema, realizando pruebas de funcionamiento para verificar su correcto accionar en el tiempo y realizando las mejoras necesarias para cumplir con los objetivos y especificaciones. El soporte técnico TIC fue prestado

por el ingeniero Darío Alejandro Riaño con la colaboración del Grupo de Investigación RadioGis.

A continuación, se introducen algunos conceptos básicos que soportan este trabajo de grado; después, cada capítulo describe el proceso para el desarrollo de cada fase y, finalmente se presentan las conclusiones más relevantes.

1. Objetivos

A continuación, se presentan los objetivos del trabajo de grado.

1.1 Objetivo General

Implementar un sistema gestor de base de datos (SGBD) para el almacenamiento, modificación y extracción de información adquirida automáticamente de la monitorización de un sistema de generación fotovoltaica (FV) integrado con vegetación.

1.2 Objetivos Específicos

- Desarrollar un sistema de adquisición de datos centralizado para un conjunto de protocolos de comunicación.
- Realizar el esquema e instancia de la base de datos centralizada para su posterior uso.
- Implementar un SGBD para la creación de la base de datos centralizada.

2. Consideraciones generales

Este capítulo describe el sistema GRIPV, el sistema de medición de variables eléctricas y aborda conceptos con respecto a bases de datos y herramientas de programación utilizadas para desarrollar este trabajo de investigación.

2.1. Sistema GRIPV

Un sistema GRIPV (Green Building Integrated Photovoltaics) consiste en la integración de techos verdes y sistemas FV, dos tecnologías que contribuyen al desarrollo sostenible. El primero, es usado para el mejoramiento térmico y aprovechamiento de aguas pluviales en edificaciones; el segundo, para la generación de energía renovable. La combinación de estas dos tecnologías busca aprovechar los efectos de enfriamiento y sombreado, ya que la evapotranspiración de las plantas del techo verde permite mejorar la eficiencia de los paneles FV y éstos protegen las plantas de la excesiva exposición al sol y la evaporación, mejorando así su crecimiento (Hui & Chan, 2011). Una imagen del sistema GRIPV de la terraza superior del Edificio de Ingeniería Eléctrica se presenta en la Figura 1.



Figura 1: Sistema GRIPV, terraza superior del Edificio de Ingeniería Eléctrica

2.2 Sistema de medición de variables eléctricas

El punto de partida de este trabajo de grado es un sistema de medición de variables eléctricas conformado por seis equipos. Para la monitorización del comportamiento eléctrico del edificio, se cuenta con dos AcuRev 2020 (Figura 4) y un Acuvim IIW (Figura 3) instalados en la zona de tableros; inicialmente los tres equipos se encontraban conectados a un mismo puerto RS485 ubicado también en la zona de tableros. El sistema FV es monitorizado por dos AcuRev 2020 instalados en la terraza superior, y un Acuvim IIR instalado en el cuarto técnico del piso 4; los primeros se encontraban conectados inicialmente a un puerto RS485 dispuesto en el cuarto técnico del piso 5, y el segundo a otro puerto RS485 ubicado en el mismo espacio.

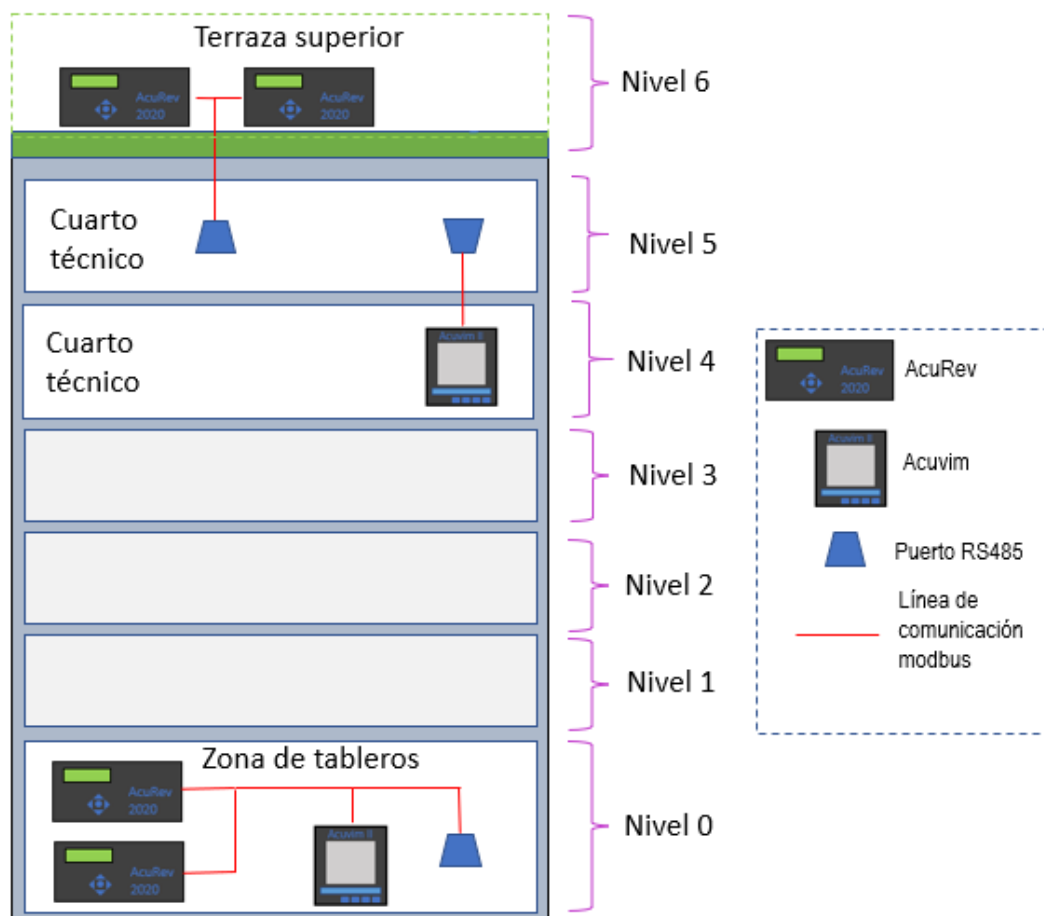


Figura 2: Ubicación inicial de los elementos del sistema de medición en el Edificio de Ingeniería Eléctrica.

La Figura 2 representa la ubicación inicial de los medidores y puertos RS485 en el Edificio de Ingeniería Eléctrica, y la conexión de las tres redes modbus que conforman el sistema de medición de variables eléctricas.

2.2.1 Acuvim IIW. Es un medidor digital de potencia y energía fabricado por ACCUENERGY que proporciona datos en tiempo real mediante una pantalla LCD o a través del puerto de comunicación. Puede medir aproximadamente 400 parámetros, tiene hasta tres tablas de registros con capacidad para almacenar la mayoría de los valores capturados por el equipo. La memoria interna disponible es de 16MB y el tamaño de cada tabla es flexible. Todas las entradas en el registro tienen su correspondiente fecha y hora. Tiene múltiples puertos de comunicación como Ethernet y RS485 y diferentes protocolos (Modbus RTU, DNP 3.0, BACnet IP, BACnet MS/TP) (Corp, 2014)



Figura 3: Acuvim II. (Corp, 2014)

2.2.2. Acurrrev 2020. Este equipo permite la medición de energía de múltiples circuitos, calcula parámetros de potencia, demanda y tiempo de uso, entre otros. Cuenta con 18 canales para 18 circuitos monofásicos o 6 circuitos trifásicos. La memoria interna disponible es de 8MB (Power, Meter, & Certified, 2000).

El AcuRev (Figura 4) también es fabricado por ACCUENERGY y comparte varias características del Acuvim, siendo de especial interés las pertinentes a la comunicación (tipos de puertos y protocolos).



Figura 4: AcuRev 2020. (Power, Meter, & Certified, 2000)

2.3 Bases de datos y sistemas de gestión de bases de datos

. Hace referencia a la evaluación e interpretación de las implicaciones de los resultados que arrojó el estudio. Se enfatiza en las consecuencias teóricas de los resultados y la validez de las conclusiones.

Una base de datos (BD) es un conjunto de datos relacionados entre sí, organizados y estructurados, con información referente a algo. A continuación, se presentan conceptos claves para el desarrollo de la base de datos implementada en este trabajo de grado.

Un sistema de gestión de base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la BD. Entre los servicios que brindan se encuentran:

- Creación y definición de la base de datos: Especificación de estructura, tipo de datos, restricciones y relaciones entre ellos, mediante lenguajes de definición de datos.
- Manipulación de datos realizando consultas, inserciones y actualizaciones.
- Acceso controlado a los datos mediante mecanismos de seguridad de acceso a los usuarios.
- Conservación de la integridad y consistencia de los datos evitando que estos sean perjudicados por cambios no autorizados
- Acceso compartido a la BD controlando la interacción entre usuarios concurrentes
- Mecanismos de copias de respaldo y recuperación para restablecer la información en caso de fallos en el sistema (Ramos Martín & Ramos Martín, n.d.).

2.4. Herramientas de programación

A continuación, se presentan algunas de las herramientas utilizadas para el desarrollo de las rutinas y aplicativos.

2.4.1. Lenguaje de programación. Puede definirse como lenguaje de programación a un conjunto de signos para describir computaciones en una forma legible tanto para la máquina como para el ser humano (Cortazar Bellas, Martinez Ananue, & Fresno Fernandez, n.d.).

Se utilizó JAVA, debido a que es un lenguaje de programación de alto nivel orientado a objetos. Permite el desarrollo de programas capaces de ejecutarse en diversas plataformas hardware y software, principalmente dispositivos electrónicos embebidos. Fue diseñado para escribir y compilar una sola vez en una plataforma, y ejecutarse en cualquier otra sin necesidad de tener que modificar el código fuente, JAVA sigue el paradigma de programación orientada a objetos, el más utilizado en la actualidad, debido a que facilita todo el ciclo de vida del software, desde el análisis y diseño hasta el mantenimiento (Garrido Avaenza, 2015).

2.4.2. Programación orientada a objetos. Se usó la programación orientada a objetos que consiste en diseñar aplicaciones y programas de computador simulando el mundo real, a partir de la definición de objetos y sus interacciones, buscando abstraer algunas características de sistemas naturales complejos como son:

- Atributos: características del objeto.
- Métodos: comportamiento del objeto.
- Clase: define propiedades y comportamientos de un tipo de objetos
- Herencia: comportamientos comunes entre objetos relacionados para hallar relaciones de especialización y generalización de comportamientos (Vargas Castillo, n.d.).

2.4.3. Entorno de Desarrollo Integrado (EDI). Entorno de Desarrollo Integrado (EDI) o IDE por sus siglas en inglés (Integrated Development Environment), es un software que proporciona herramientas y características que facilitan el trabajo de los desarrolladores, normalmente incluye un editor, herramientas de compilación y depurador. El desarrollo de programas JAVA requiere la utilización de un IDE, se usó NetBeans, un entorno gratuito para desarrollar programas estándar y applets. Facilita la realización de aplicaciones, permitiendo la

compilación, empaquetado, distribución y ejecución dentro del mismo entorno. Está disponible para las principales plataformas, Microsoft Windows, Linux y Mac (Garrido Avaenza, 2015).

3. Desarrollo del sistema de adquisición de datos

La adquisición centralizada de los datos obtenidos por varios equipos de medición busca facilitar el direccionamiento de la información y su ingreso a la base de datos. Este capítulo describe el desarrollo del sistema de adquisición de datos y sus características.

3.1. Comunicación a partir del protocolo modbus

Un protocolo de comunicación instauro un conjunto de reglas y formatos de mensajes para transmitir información entre emisores y receptores. Se seleccionó Modbus debido a las características que comparten el Acuvim II y el AcuRev 2020, ya que disponen de puerto RS485 y la capacidad de comunicarse por medio de este protocolo.

Modbus es un protocolo serial donde un maestro puede conectarse simultáneamente con un máximo de 247 esclavos; cada uno de éstos se identifica mediante una dirección única, su implementación es posible sobre redes basadas en Ethernet, RS485 y RS-232 (Ver Figura 5).

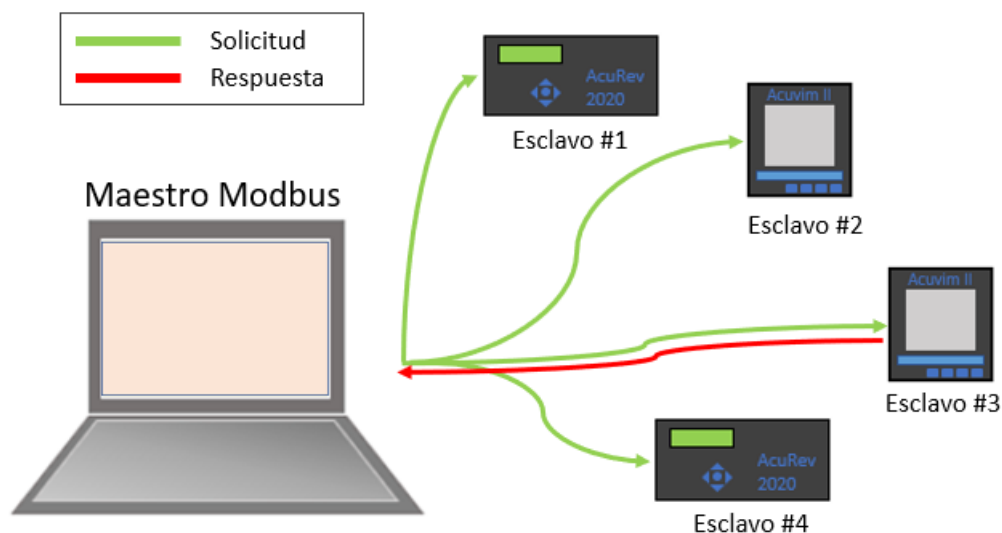


Figura 5: Red de comunicación Modbus

Para establecer comunicación, el maestro envía una secuencia de datos que especifica a qué esclavo se dirige, la función solicitada y los registros sobre los cuales ésta se ejecuta. Como muestra la Figura 5, la solicitud es recibida por todos los dispositivos conectados a la red modbus, pero únicamente contestada por aquel al que se dirige. La Tabla 1 muestra la estructura de la trama de datos para las solicitudes

Tabla 1.

Estructura de la trama de datos para consultas en el protocolo Modbus.

Dirección del esclavo	Función	Registro de inicio	Número de registros	CRC
8 bits	8 bits	16 bits	16 bits	16 bits

Dirección del esclavo: Es la primera parte de la estructura; está compuesta por un byte, su rango de valores decimales es de 0 a 247. Para dirigirse a un esclavo, el maestro coloca su dirección en este campo, cuando el esclavo envía su respuesta, coloca su propia dirección para identificarla.

Función: El código de función le indica al esclavo que tipo de acción realizar; la Tabla 2 presenta los códigos y las funciones más utilizadas (Monge Gómez, n.d.). El único código empleado para desarrollar este trabajo es 3, que corresponde a leer registros.

Tabla 2.

Códigos de función más utilizados en el protocolo Modbus.

<i>Código Decimal</i>	<i>Función</i>	<i>Tipo de dato</i>
1	Leer estado de marcas y salidas digitales	Bit
2	Leer estado de entradas digitales	Bit
3	Leer registros	16 bits
4	Leer entradas analógicas	16 bits
5	Forzar valor de una salida digital	Bit
16	Escribir en un registro	16 bits

Registro de inicio: En los equipos Modbus, las mediciones de cada parámetro son almacenadas en uno o más registros; en este campo se indica a partir de cual registro se inicia la lectura.

Número de registros: En este campo se indica el número de registros a leer, aunque la rutina desarrollada hace una solicitud por cada variable, hay variables almacenadas en más de un registro. La posibilidad de leer varios registros consecutivos es una característica valiosa del protocolo Modbus si se busca reducir el tiempo de la comunicación; sin embargo, es importante verificar la máxima cantidad de registros que pueden leerse en una solicitud, ya que esto varía en cada equipo.

CRC: La verificación por redundancia cíclica (Cyclic Redundancy Check) es el método utilizado para evitar que los dispositivos respondan a datos distorsionados durante la transmisión, mejorando la efectividad y eficiencia del sistema. El valor del CRC se calcula por el dispositivo transmisor y se anexa al mensaje, el receptor recalcula el CRC y lo compara con el valor en este campo; si los valores no son iguales se anuncia un error (Power et al., 2000).

La Tabla 3 contiene un ejemplo de consulta mediante protocolo modbus; en este caso se solicita al equipo número 1 (Acuvim), leer el registro 16384, el sistema de numeración usado es el hexadecimal.

Tabla 3.

Ejemplo de consulta mediante protocolo Modbus.

Dirección del esclavo	Función	Registro de inicio	Número de registros	CRC
01	03	4000	1	91CA

La respuesta lleva como encabezado la dirección del esclavo y la función, seguidas por la cantidad de datos, la información extraída de los registros y finalmente el CRC. La Tabla 4 muestra la estructura de la trama de datos para la respuesta. N representa el doble de la cantidad de registros solicitados.

Tabla 4.

Estructura de la trama de datos de las respuestas en el protocolo Modbus.

Dirección del esclavo	Función	Cantidad de datos	Datos	CRC
8bits	8bits	8 bits	Nx8bits	16bits

Para verificar el funcionamiento de los puertos RS485 y establecer la cantidad de registros que puede leer cada equipo en una solicitud, se realizó una primera conexión vía Modbus usando el software Hércules SETUP. Este programa tiene entre sus funciones, actuar como terminal para

puerto serie con soporte para trabajar con puertos virtuales, la aplicación de escritorio puede descargarse gratuitamente.

La ventana de comunicación serial del software (Figura 6), tiene tres líneas de macros en las cuales se escriben las tramas de datos del protocolo Modbus en formato ASCII o HEX, también deben seleccionarse parámetros básicos como el nombre del puerto de destino, la velocidad en baudios y la paridad. Para poder enviar un mensaje es necesario abrir el puerto mediante el botón destinado para esto.

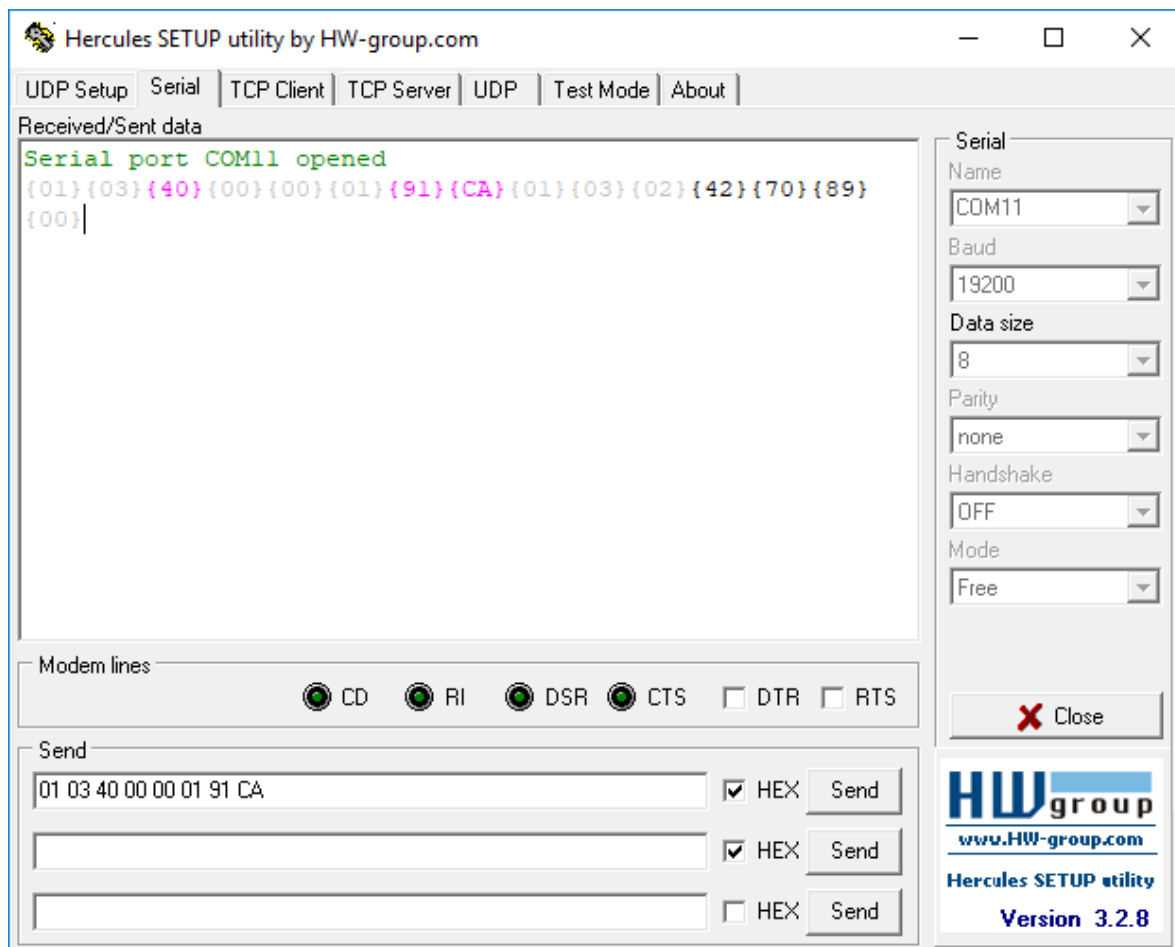


Figura 6. Ejemplo de comunicación usando el software Hercules SETUP.

En el ejemplo mostrado por la Figura 6, la trama de datos enviada corresponde a la presentada en la Tabla 3, que solicita al esclavo número 1 (Acuvim); leer el registro 16384 (4000H), éste almacena la frecuencia. En la ventana de datos enviados y recibidos, se visualiza la solicitud realizada seguida por la respuesta del equipo (Tabla 5).

Tabla 5.

Ejemplo de respuesta mediante protocolo Modbus.

Dirección del esclavo	Función	Cantidad de datos	Datos	CRC
01	03	02	42 70	89 00

El número hexadecimal 4270 corresponde a la secuencia binaria 0100001001110000; el resultado de la conversión de esta secuencia a flotante es 60, que corresponde al valor esperado de la frecuencia.

3.2. Rutina para adquisición de datos

La rutina desarrollada automatiza el procedimiento explicado en el numeral anterior, haciendo consultas de cada uno de los parámetros; además, hace las conversiones necesarias para entregar los datos en formato decimal.

Como se mencionó, para establecer comunicación efectiva mediante protocolo Modbus, es necesario identificar los registros correspondientes a cada variable. Como entrada de la rutina, se ingresa para cada equipo un archivo en formato csv, especificando para cada parámetro el número de registro de inicio, el tipo de variable y un factor por el cual debe multiplicarse la respuesta conforme a las especificaciones del fabricante.

Número de registro de inicio: Corresponde al espacio de memoria donde el equipo almacena un parámetro o la primera parte de éste.

Conversión: Para aprovechar al máximo la capacidad del protocolo de comunicación y aumentar el número de cifras significativas, el valor entregado por el equipo no siempre corresponde a la medición real; en muchos casos es necesario aplicar una multiplicación, estas son especificaciones del fabricante y varían de acuerdo con el tipo de equipo y a la variable medida.

Tipo de variable: Esta característica permite identificar la cantidad de registros que ocupa un parámetro cuando se hacen las consultas y la forma en que debe realizarse la conversión de la secuencia binaria cuando el equipo responde.

La Figura 7 presenta una vista del documento de entrada en el caso del Acuvim.

	A	B	C	D
1	Nombre variable	#registro	conversion	tipo
2	Frequency	16384	a*1	float
3	Phase voltage V1	16386	a*1	float
4	Phase voltage V2	16388	a*1	float
5	Phase voltage V3	16390	a*1	float
6	Average voltage Vavg	16392	a*1	float
7	Line voltage V12	16394	a*1	float
8	Line voltage V23	16396	a*1	float
9	Line voltage V31	16398	a*1	float
10	Average line voltage Vlavg	16400	a*1	float
11	Current I1	16402	a*1	float
12	Current I2	16404	a*1	float
13	Current I3	16406	a*1	float
14	Average current lavg	16408	a*1	float
15	Neutral current In	16410	a*1	float
16	Phase A power Pa	16412	a*0.001	float
17	Phase B power Pb	16414	a*0.001	float
18	Phase C power Pc	16416	a*0.001	float
19	System power Psum	16418	a*0.001	float
20	Phase A reactive power Qa	16420	a*0.001	float

Figura 7: Vista del documento de entrada del Acuvim

Los tipos de variables que manejan los equipos son:

- Word: entero sin signo de 16 bits, utiliza dos registros. El rango de datos es 0 a 65535.
- Int: entero con signo de 16 bits, utiliza dos registros. El rango de datos es de 32768 a 32767.
- Dword: Entero sin signo de 32 bits, utiliza 4 registros. El rango de datos es de 0 a 4294967295

- Float: Punto flotante de precisión simple, utiliza 2 registros. El rango de datos es 0.0 a 3.402823×10^{38} (Corp, 2014).

La mayor parte de la información incluida en el documento de entrada puede extraerse de los manuales de los equipos; sin embargo, el multiplicador no está explícito para todas las variables; por lo tanto, algunos de estos valores se calcularon experimentalmente.

Otros datos que deben ingresarse son el puerto de comunicación, el número de lecturas a realizar, la dirección del equipo, el tipo (Acuvim o AcuRev) y el tiempo entre lecturas. Esta información puede ser suministrada mediante una interfaz gráfica de usuario (Figura 8).

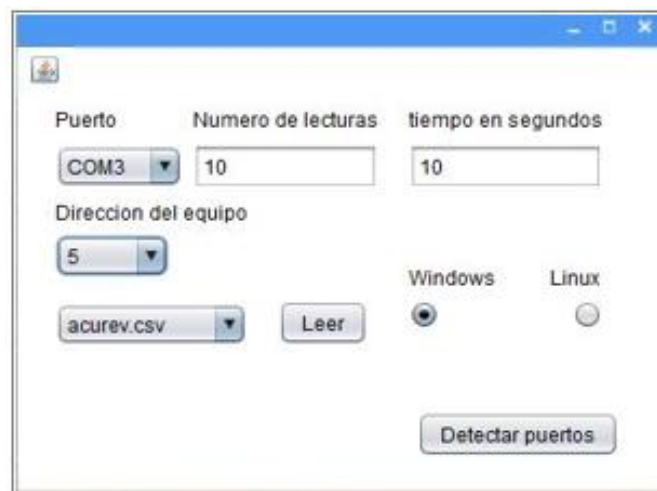


Figura 8: Interfaz gráfica de usuario.

En una lectura se realizan tantas consultas como parámetros contenga el documento de entrada; cada vez que se termina este proceso, el programa comunica el número de la lectura realizada y el tiempo que tardó. Los valores de cada parámetro, calculados a partir de la

respuesta del equipo, son almacenados en un archivo llamado Historial.csv y, finalmente, hace una pausa hasta que se cumple el tiempo indicado por el usuario para realizar una nueva lectura.

En la Figura 9, las flechas verdes representan los parámetros de entrada que permiten realizar las consultas; mientras las flechas rojas hacen referencia a las salidas generadas a partir de la respuesta del equipo.

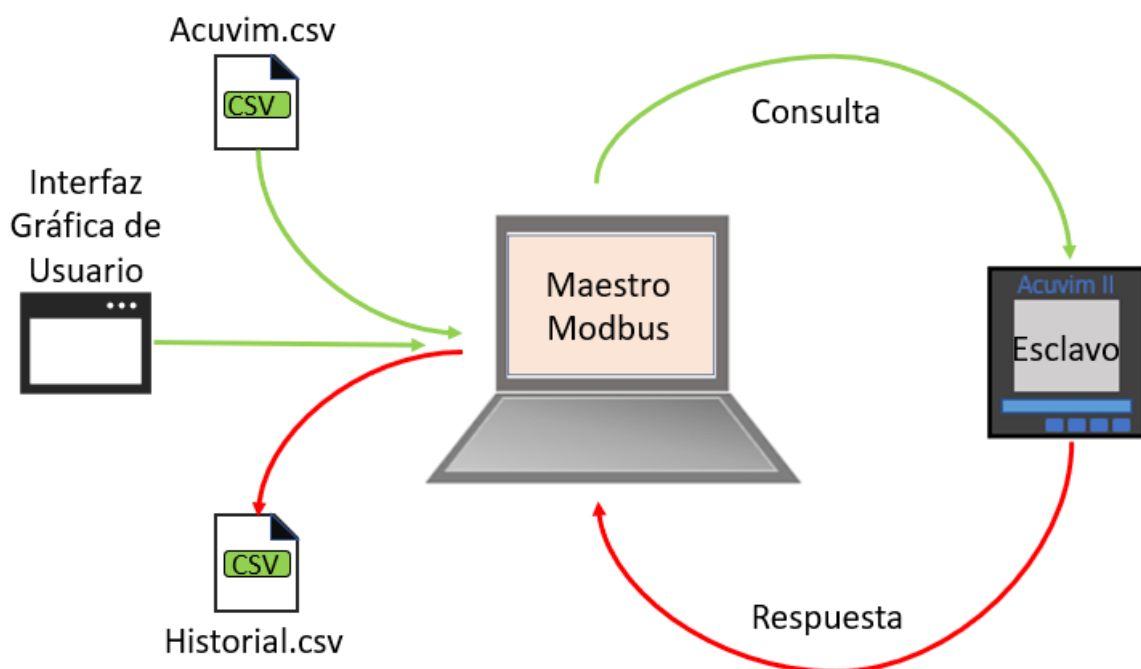


Figura 9: Entradas y salidas de la rutina

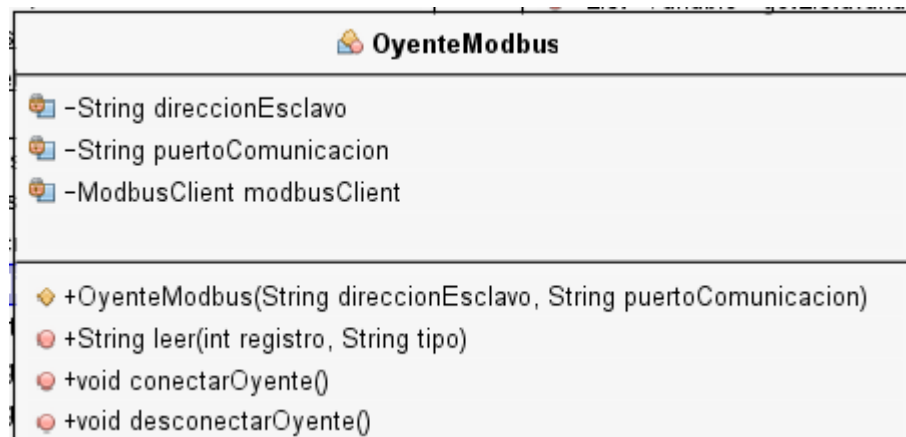
La rutina finaliza una vez se realiza el total de lecturas indicadas por el usuario; como salida, entrega el archivo con los valores de cada parámetro, indicando la fecha y hora de las lecturas, se presenta un ejemplo en la Figura 10. Como se mencionó en el Capítulo 2, la rutina está escrita en JAVA.

	A	B	C	D	E	F	G
	#lectura	hora	Frequency	Phase voltage V1	Phase voltage V2	Phase voltage V3	Average voltage Vavg
1	1	21/05/2019 10:00	60,000000	124,439987	124,622520	124,299599	124,454041
2	2	21/05/2019 10:10	60,029999	124,734940	124,961411	124,573341	124,756554
3	3	21/05/2019 10:20	60,020000	124,784119	124,906006	124,470421	124,720177
4	4	21/05/2019 10:30	60,009998	125,050766	125,130005	124,783813	124,988197
5	5	21/05/2019 10:40	59,930000	124,805588	124,928185	124,614365	124,782715
6	6	21/05/2019 10:50	59,939999	124,586105	124,704567	124,316002	124,535561
7	7	21/05/2019 11:00	60,009998	124,340843	124,520470	124,081711	124,314346
8	8	21/05/2019 11:10	59,930000	124,115982	124,200058	123,835159	124,050400
9	9	21/05/2019 11:20	59,970001	123,868996	124,039894	123,672340	123,860413
10	10	21/05/2019 11:30	59,950001	124,167770	124,323471	124,000931	124,164062
11	11	21/05/2019 11:40	59,990002	123,821594	123,982361	123,610268	123,804733
12	12	21/05/2019 11:50	59,970001	123,973526	124,099770	123,775070	123,949448
13	13	21/05/2019 12:00	60,029999	123,884727	124,110985	123,806030	123,933914
14	14	21/05/2019 12:10	60,000000	123,973145	124,176254	123,873337	124,007584
15	15	21/05/2019 12:20	59,950001	124,012024	124,095657	123,805130	123,970932
16	16	21/05/2019 12:30	59,950000	124,001276	124,040005	123,840105	123,983503

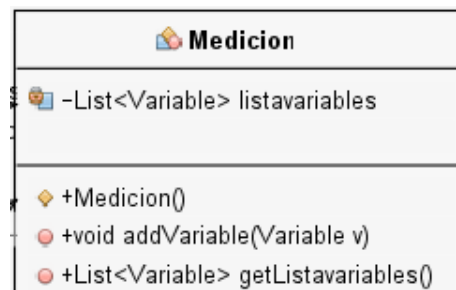
Figura 10: Vista del documento de salida de la rutina.

3.2.1. Elementos de la rutina. A continuación, se mencionan algunos de los elementos (clases, métodos y objetos)⁵ que conforman la rutina:

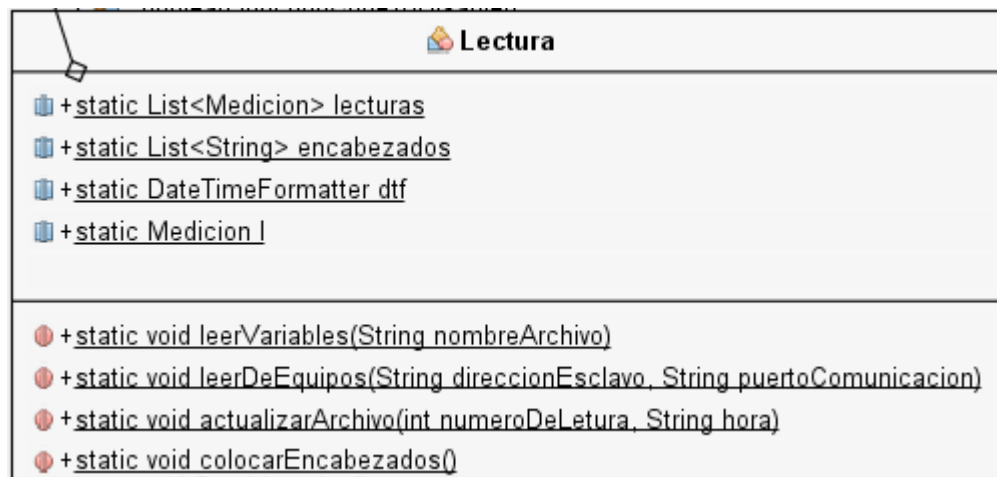
⁵ La información sobre los elementos de la rutina fue proporcionada por el Ingeniero Alejandro Riaño



a. Oyente



b. Medición



c. Lectura

Figura 11: Visualización de algunos elementos de la rutina en el diagrama.

- Oyente Modbus: Se comunica con los equipos y utiliza el método llamado lectura para solicitar la dirección de los registros y el tipo de variable, realiza la conversión necesaria y obtiene el valor de cada una (Ver Figura 11.a).
- Medición: Almacena la totalidad de los datos obtenidos en una lectura (Ver Figura 11.b).
- Lectura: Accede a los archivos que contienen las variables correspondientes a cada equipo, se comunica con éstos y guarda los valores mediante el objeto medición. (Ver Figura 11.c).
- Demo: Accede a una calculadora y multiplica el valor de la medición por el factor indicado en el archivo de entrada, con el fin de entregar el valor real.

3.2.2. EasyModbus. La Librería EasyModbus facilita la integración independiente de la plataforma del protocolo Modbus con la biblioteca JAVA. Puede descargarse fácilmente, es gratuita y permite modificaciones (Rossmann-Engineering, n.d.). Para desarrollar la rutina de adquisición de datos se hizo una adaptación de esta librería.

3.3. Pruebas de funcionamiento de la rutina de adquisición de datos

El objetivo de las pruebas es verificar la validez de la información obtenida mediante la rutina, para esto se realizan varias tomas de datos sincronizadas con el almacenamiento en memoria de los equipos y se comparan los resultados almacenados mediante protocolo Modbus con aquellos descargados directamente de los medidores.

La primera versión de la rutina se configuró para tomar datos cada 10 minutos de un Acuvim y un AcuRev durante una semana. Al finalizar la prueba se encontró que los datos no podían compararse debido a que había un desfase total de aproximadamente 3 horas para el Acuvim y 5 horas para el AcuRev. Esto se debe a que el programa corría cada 10 minutos una vez terminaba la lectura, lo cual tardaba entre 3 y 7 segundos, según el equipo, y este tiempo se acumula causando el desfase.

Para corregir el error, se configuró la rutina para restar el tiempo de la lectura anterior al intervalo entre lecturas. Esta corrección mejoró ampliamente la sincronización del equipo; sin embargo, en las pruebas siguientes se registró un desfase de uno o dos minutos diarios, causado porque el programa tomaba un valor truncado de los tiempos de lectura, por ejemplo, si esta tardaba 3,598 segundos, el programa corría a los 9 minutos con 57 segundos, por lo cual resultó de suma importancia tener en cuenta los milisegundos. Solucionado el error de sincronización, fue posible comparar los datos. Debe tenerse en cuenta que el programa consulta las variables una a una, y el equipo las guarda en memoria instantáneamente; por lo tanto, hay segundos de diferencia para los últimos parámetros consultados.

Para comparar los datos se realizaron gráficas de datos Modbus con respecto a datos descargados, se calcula el índice de correlación del conjunto de datos linealizados para medir el grado de relación y la pendiente de la recta se aproxima al multiplicador, lo cual permite calcular este factor si no está especificado en el manual. La Figura 12 presenta las gráficas realizadas para dos parámetros; Voltaje AN (Figura 12.a) y Voltaje BN (Figura 12.b).

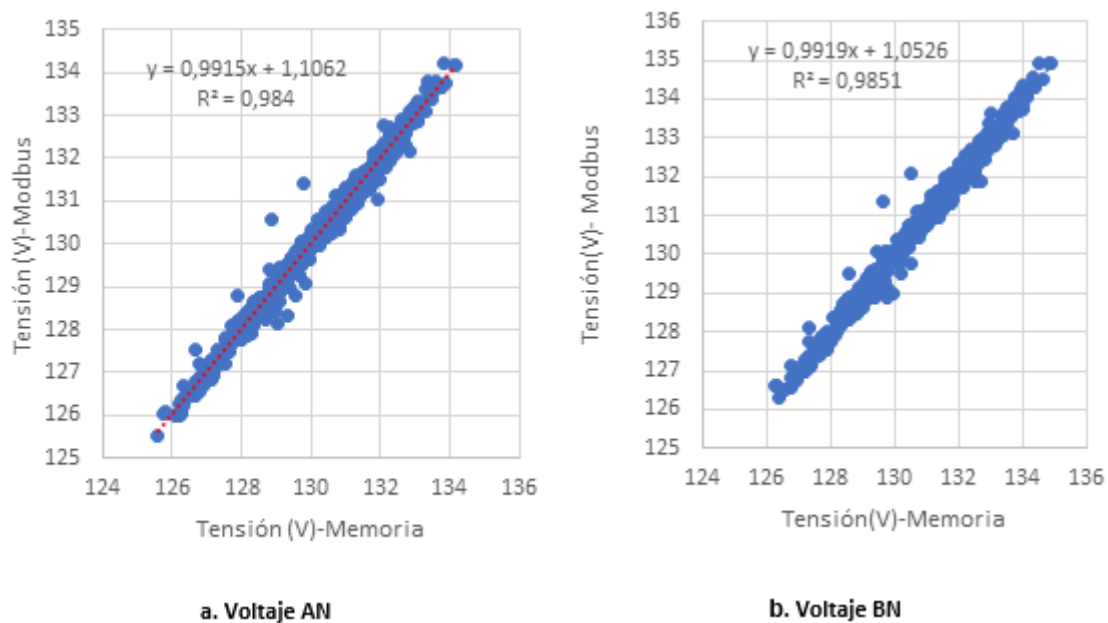


Figura 12: Gráficas de datos Modbus vs datos descargados para dos parámetros del Acuvim

Las tablas a continuación resumen los resultados de la primera comparación de datos, los porcentajes de error entre 5% y 10% se encuentran resaltados en amarillo, y aquellos superiores a 10% se resaltan con color rojo.

Tabla 6.

Resultados de la comparación de datos extraídos por el AcuRev.

Parámetro	r ²	Error porcentual	Parámetro	r ²	Error porcentual
Frequency	0.066	74.35%	PwrFactorA	0.980	1.00%
IA	0.990	0.52%	PwrFactorA(CH101)	0.983	0.84%
IA(CH101)	1	0.00%	PwrFactorA(CH103)	0.889	5.70%
IA(CH102)	1	0.00%	PwrFactorA(CH104)	1	0.00%
IA(CH103)	0.990	0.50%	PwrFactorA(CH105)	0.543	26.25%
IA(CH104)	1	0.00%	PwrFactorA(CH106)	0.992	0.40%

IA(CH105)	0.947	2.69%	PwrFactorA(CH107)	1	0.00%
IA(CH106)	0.996	0.19%	PwrFactorA(CH108)	0.705	16.03%
IA(CH107)	1	0.00%	PwrFactorA(CH109)	0.989	0.55%
IA(CH108)	0.950	2.53%	PwrFactorA(CH110)	0.998	0.08%
IA(CH109)	1	0.00%	PwrFactorA(CH111)	0.432	34.24%
IA(CH110)	1	0.00%	PwrFactorA(CH112)	0.999	0.04%
IA(CH111)	0.999	0.04%	PwrFactorA(CH113)	0.996	0.18%
IA(CH112)	0.986	0.68%	PwrFactorA(CH114)	0.943	2.88%
IA(CH113)	1	0.01%	PwrFactorA(CH115)	0.999	0.03%
IA(CH114)	1	0.00%	PwrFactorA(CH116)	0.636	20.22%
IA(CH115)	0.986	0.68%	PwrFactorA(CH117)	0.473	31.23%
IA(CH116)	0.999	0.06%	PwrFactorA(CH118)	0.999	0.04%
IA(CH117)	0.998	0.12%	PwrFactorTotal	0.983	0.85%
IA(CH118)	0.987	0.63%	PwrFactorTotal(CH301/201)	1	0.00%
IAverage	0.989	0.53%	PwrFactorTotal(CH302/202)	0.641	19.91%
IB	0.990	0.49%	PwrFactorTotal(CH303/203)	0.966	1.69%
IC	0.981	0.93%	PwrFactorTotal(CH304/204)	0.995	0.24%
PwrFactorB	0.983	0.87%	PwrFactorTotal(CH305/205)	0.816	9.67%
PwrFactorC	0.978	1.12%	PwrFactorTotal(CH306/206)	0.999	0.03%
VAA	0.990	0.50%	VATotal(CH301/201)	1	0.00%
VAA(CH101)	1	0.00%	VATotal(CH302/202)	0.985	0.74%
VAA(CH102)	0.988	0.62%	VATotal(CH303/203)	0.999	0.06%
VAA(CH103)	0.990	0.51%	VATotal(CH304/204)	1	0.00%
VAA(CH104)	1	0.00%	VATotal(CH305/205)	0.999	0.03%
VAA(CH105)	0.921	4.04%	VATotal(CH306/206)	0.990	0.52%
VAA(CH106)	0.996	0.20%	VARTotal(CH301/201)	0.993	0.36%
VAA(CH107)	1	0.00%	VARTotal(CH302/202)	0.987	0.64%
VAA(CH108)	0.949	2.58%	VARTotal(CH303/203)	0.984	0.80%
VAA(CH109)	1	0.00%	VARTotal(CH304/204)	0.997	0.16%
VAA(CH110)	1	0.00%	VARTotal(CH305/205)	0.995	0.23%
VAA(CH111)	0.999	0.04%	VARTotal(CH306/206)	0.971	1.44%

VAA(CH112)	0.987	0.66%	WattTotal	0.968	1.60%
VAA(CH113)	1	0.00%	WattTotal(CH301/201)	1	0.00%
VAA(CH114)	1	0.00%	WattTotal(CH302/202)	0.988	0.61%
VAA(CH115)	0.987	0.67%	WattTotal(CH303/203)	0.999	0.07%
VAA(CH116)	0.999	0.04%	WattTotal(CH304/204)	1	0.00%
VAA(CH117)	0.999	0.06%	WattTotal(CH305/205)	1	0.00%
VAA(CH118)	0.988	0.62%	WattTotal(CH306/206)	0.994	0.31%
VAB	0.991	0.47%	WattA(CH101)	1	0.00%
VAC	0.982	0.89%	WattA(CH102)	1	0.00%
VARA	0.961	1.96%	WattA(CH103)	0.990	0.49%
VARA(CH101)	1	0.00%	WattA(CH104)	1	0.00%
VARA(CH102)	0.979	1.05%	WattA(CH105)	0.928	3.68%
VARA(CH103)	0.953	2.38%	WattA(CH106)	0.996	0.22%
VARA(CH104)	1	0.00%	WattA(CH107)	1	0.00%
VARA(CH105)	0.931	3.51%	WattA(CH108)	0.955	2.25%
VARA(CH106)	0.998	0.08%	WattA(CH109)	0.993	0.37%
VARA(CH107)	0.999	0.03%	WattA(CH110)	0.999	0.07%
VARA(CH108)	0.897	5.28%	WattA(CH111)	0.999	0.06%
VARA(CH109)	0.995	0.25%	WattA(CH112)	0.991	0.47%
VARA(CH110)	0.990	0.52%	WattA(CH113)	1	0.00%
VARA(CH111)	0.996	0.18%	WattA(CH114)	0.999	0.04%
VARA(CH112)	0.977	1.13%	WattA(CH115)	0.991	0.44%
VARA(CH113)	0.997	0.13%	WattA(CH116)	0.996	0.18%
VARA(CH114)	0.989	0.57%	WattA(CH117)	0.997	0.14%
VARA(CH115)	0.877	6.34%	WattA(CH118)	0.992	0.41%
VARA(CH116)	0.993	0.35%	WattB	0.968	1.60%
VARA(CH117)	0.991	0.45%	WattC	0.961	1.97%
VARA(CH118)	0.986	0.68%	VoltsLLAverage	0.989	0.56%
VARB	0.944	2.86%	VoltsLNAverage	0.986	0.68%
VARC	0.930	3.57%	H1THDVA	0.988	0.61%
VARTotal	0.972	1.42%	H1THDVB	0.985	0.76%
VoltsAB	0.987	0.65%	H1THDVC	0.986	0.70%

VoltsAN	0.987	0.66%	H1UnbalanceV	0.734	14.32%
VoltsBC	0.988	0.61%	WattA	0.979	1.07%
VoltsBN	0.988	0.61%	PwrFactorA(CH102)	0.987	0.65%
VoltsCA	0.988	0.62%	VATotal	0.990	0.52%
VoltsCN	0.987	0.65%			

Tabla 7.

Resultados de la comparación de datos extraídos por el Acuvim.

Parámetro	r^2	Error porcentual	Parámetro	r^2	Error porcentual
Frequency	0.155	60.66%	Crest Factor V A	0.986	0.68%
I A	0.964	1.81%	Crest Factor V B	0.982	0.91%
I Average	0.964	1.81%	Crest Factor V C	0.981	0.95%
I B	0.964	1.82%	Harmonic I A 13rd	0.621	21.23%
I C	0.964	1.80%	Harmonic I A 17th	0.746	13.63%
K Factor I A	0.973	1.35%	Harmonic I A 19th	0.944	2.86%
K Factor I B	0.986	0.71%	Harmonic I A 5th	0.918	4.21%
K Factor I C	0.985	0.78%	Harmonic I B 13rd	0.548	25.97%
Pwr Factor A	1	0.01%	Harmonic I B 17th	0.913	4.43%
Pwr Factor B	1	0.01%	Harmonic I B 18th	0.937	3.20%
Pwr Factor C	1	0.01%	Harmonic I B 19th	0.937	3.20%
THD I A	0.749	13.44%	Harmonic I B 5th	0.934	3.36%
THD I B	0.87	6.73%	Harmonic I B 7th	0.743	13.83%
THD I C	0.893	5.53%	Harmonic I C 13rd	0.655	19.09%
THFF V A	0.538	26.69%	Harmonic I C 17th	0.911	4.54%
THFF V B	0.338	41.85%	Harmonic I C 19th	0.934	3.37%
THFF V C	0.453	32.69%	Harmonic I C 5th	0.941	2.99%
VA A	0.963	1.89%	Harmonic I C 7th	0.852	7.71%
VA B	0.963	1.89%	Harmonic V A 5th	0.99	0.52%
VA C	0.962	1.91%	Harmonic V A 7th	0.974	1.33%

VA Total	0.962	1.92%	Harmonic V B 5th	0.987	0.63%
VAR A	0.968	1.60%	Harmonic V B 7th	0.96	2.03%
VAR B	0.872	6.61%	Harmonic V C 5th	0.988	0.61%
VAR C	0.952	2.41%	Harmonic V C 7th	0.956	2.25%
VAR Total	0.97	1.54%	Pwr Factor Total	1	0.01%
Volts AB	0.984	0.83%	THD Volts AN/AB	0.991	0.47%
Volts AN	0.984	0.80%	THD Volts Average	0.991	0.47%
Volts BC	0.984	0.80%	THD Volts BN/CA	0.988	0.59%
Volts BN	0.985	0.75%	THD Volts CN/BC	0.989	0.56%
Volts CA	0.983	0.85%	THD I Average	0.893	5.50%
Volts CN	0.984	0.82%	Unbalance I	0.826	9.12%
Watt A	0.968	1.64%	Unbalance V	0.718	15.27%
Watt B	0.968	1.64%	Volts LL Average	0.984	0.80%
Watt C	0.968	1.63%	Volts LN Average	0.985	0.78%
Watt Total	0.968	1.63%			

Se revisaron los parámetros con altos porcentajes de error con el fin de encontrar fallas en el programa. Se concluyó que en la mayoría de los casos el error se debe a que hay cambios instantáneos en las medidas de los parámetros cuando se realiza el muestreo o lectura con protocolo Modbus, este fenómeno no se presenta en los datos descargados, puesto que la información almacenada cada 10 minutos en la memoria del equipo, corresponde al promedio de las mediciones minuto a minuto.

En el caso de la frecuencia (Figura 13), el alto porcentaje de error se debe a que su variación con respecto al tiempo no es significativa, los valores medidos siempre son muy cercanos a 60, por lo tanto, el comportamiento no es lineal.

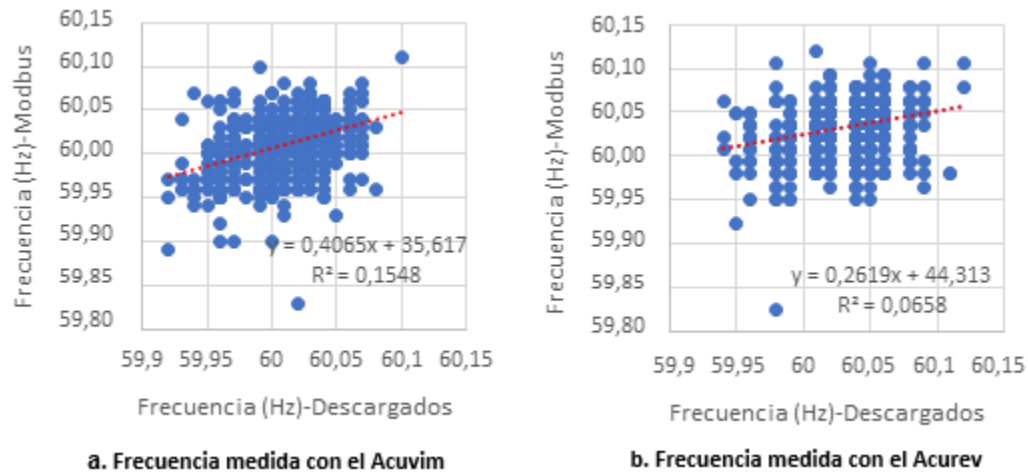


Figura 13: Gráfica de la frecuencia medida por los dos equipos

Se incluyeron más parámetros y se realizaron pruebas similares; inicialmente, los resultados no fueron del todo favorables debido a errores en la conversión de variables del tipo int, una vez se corrigió el código, los valores obtenidos fueron los esperados.

4. Diseño de la base de datos

Este capítulo describe el proceso de diseño de la base de datos, partiendo por la definición de los requerimientos y dando a conocer las herramientas y procedimientos para la elaboración de esquemas y modelos.

Existen varios modelos para construir una base de datos; se usó el modelo relacional, por su facilidad en la aplicación (Arias, 2014).

4.1. Requerimientos

1. La base de datos debe contener información básica de los equipos que conforman el sistema de adquisición de datos.
2. Cada vez que un equipo realiza una medición, deben almacenarse todos los valores indicando el momento en que se realizó la lectura y el tiempo que tardó en hacerla.

4.2. Modelo conceptual

Es el primer modelo desarrollado para crear una base de datos, su objetivo es hacer un esbozo de su estructura. Existen varias formas de crear el modelo conceptual de una base de datos; sin embargo, el más utilizado es el diagrama entidad-relación, que busca abstraer problemas del mundo real y convertir los seres, acontecimientos y cosas en entidades, estableciendo vínculos entre éstas (Arias, 2014). Los elementos que conforman este modelo y un ejemplo de su representación gráfica se muestran a continuación.

4.2.1. Entidades. Son objetos que existen en el mundo real y que debido a sus características son distinguibles de otros objetos, su representación gráfica típica se muestra en la Figura 14.A. El tipo de entidad es el esquema que tiene una entidad en la base de datos, es decir, todos los atributos y características que la definen.

- Entidad fuerte: es aquella cuya existencia no depende de otras entidades
- Entidad débil: Es aquella cuya existencia depende de otras entidades.

En la base de datos desarrollada las entidades fuertes son los equipos que conforman el sistema de adquisición de datos y las entidades débiles las mediciones.

4.2.2. Relaciones. Se trata de la forma en que las entidades actúan unas sobre otras o están asociadas entre ellas, si las entidades pueden concebirse como objetos, debe pensarse en las relaciones como verbos, se representa gráficamente como se muestra en la Figura 14.B. La relación entre las entidades de la base de datos es la acción de los equipos al realizar las mediciones.

4.2.3. Atributos. Son las propiedades o características de una entidad, la Figura 14.C muestra la representación gráfica de un atributo. A la cantidad de atributos que se relacionan entre una y otra entidad se le conoce como cardinalidad (Santamar, 1997). La Figura 14.D, indica la representación gráfica de los diferentes tipos de cardinalidad entre entidades.

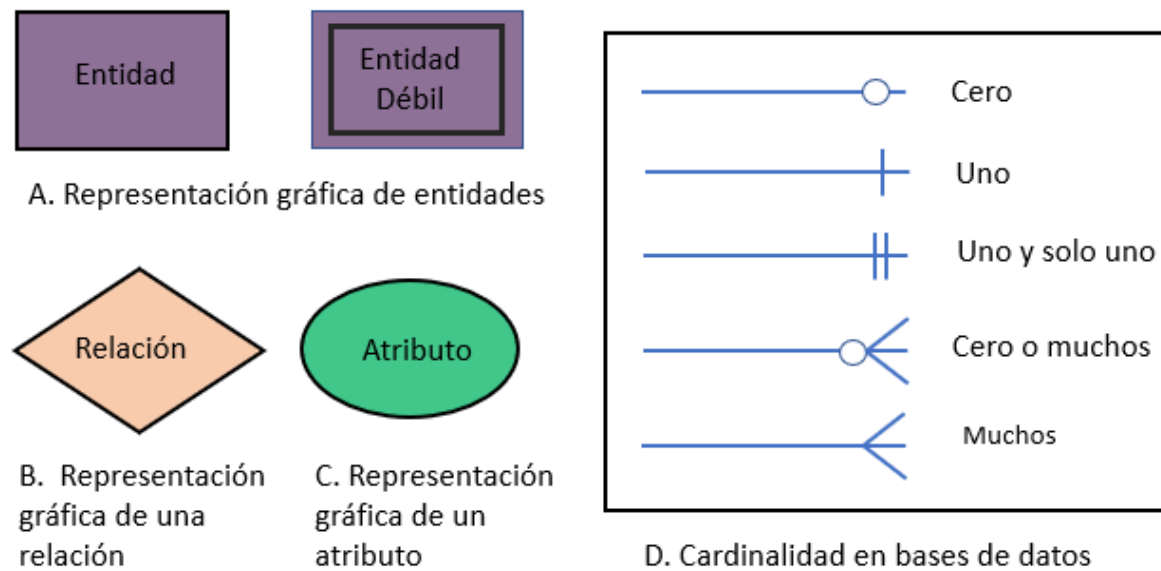


Figura 14: Elementos del modelo conceptual

4.2.4. Diagrama entidad-relación (DER). A partir de lo anterior se realiza el primer modelo de la base de datos (Figura 15). Por ser un sistema entidad-relación, se representa su contenido y estructura independientemente del SGBD a utilizar.

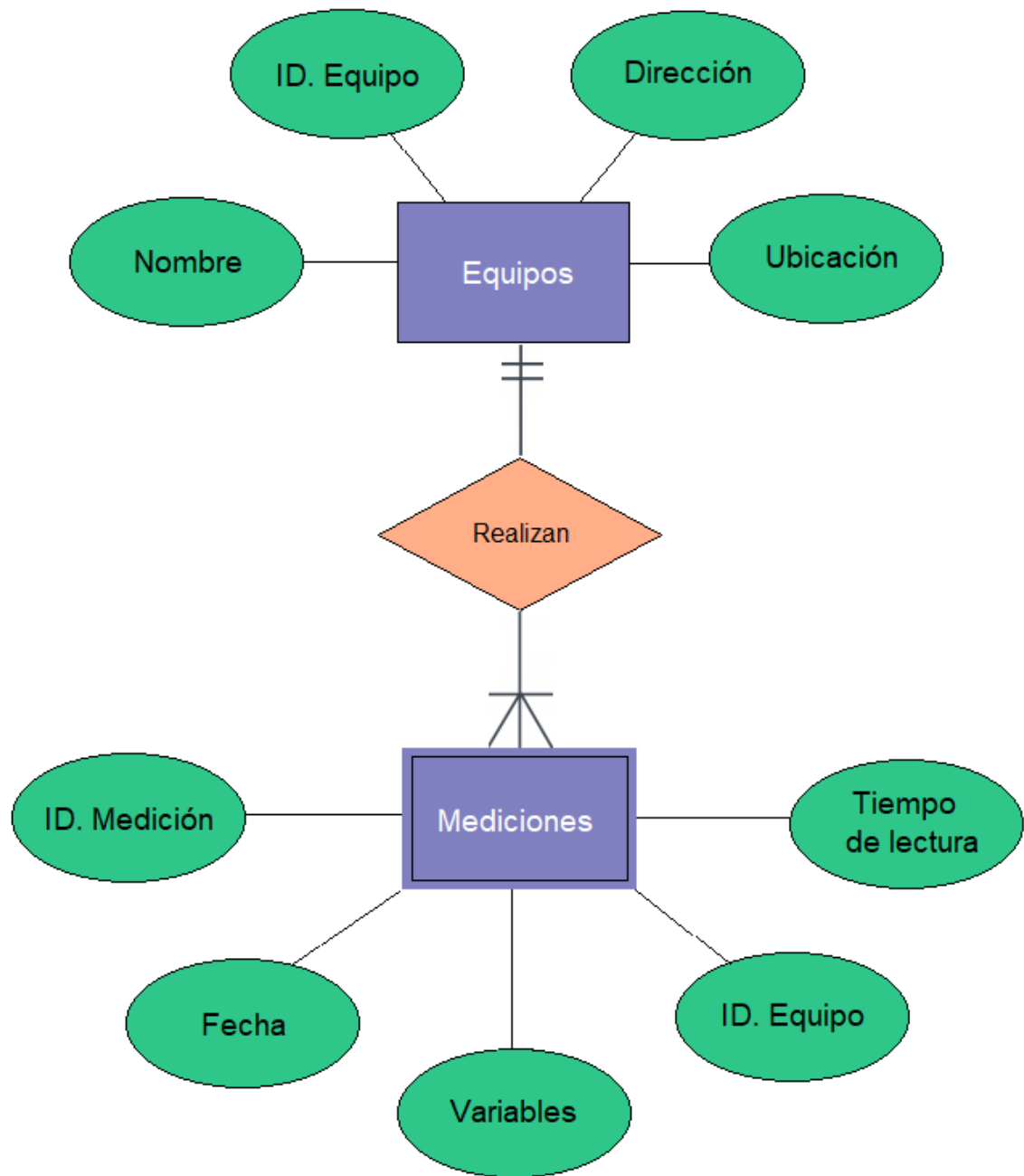


Figura 15: Diagrama entidad-relación de la base de datos

4.3. Modelo relacional

Este modelo organiza los datos en tablas y establece relaciones entre ellas.

4.3.1. Definiciones básicas. A continuación, se definen algunos elementos del modelo relacional.

- Relación: Tabla con columnas y filas
- Atributo: Nombre de una columna
- Dominio: Conjunto de valores legales del atributo
- Tupla: fila de una relación

En la Figura 16 se especifica cada uno de los conceptos anteriores para la tabla equipos, identificar correctamente estos elementos facilita el ejercicio de pasar del diagrama entidad-relación al modelo relacional.

EQUIPOS			
ID. EQUIPO	NOMBRE	DIRECCIÓN	UBICACIÓN
SAV1	Acuvim IIW	1	Zona de tableros
SAR2	AcuRev 2020	2	Zona de tableros
SAR3	AcuRev 2020	3	Zona de tableros
5PAR4	AcuRev 2020	4	Terraza superior
5PAR5	AcuRev 2020	5	Terraza superior
5PAV6	Acuvim IIR	6	Cuarto técnico piso 4

Figura 16: Conceptos aplicados a la tabla Equipos de la base de datos

4.3.2. Llaves relacionales. En el modelo relacional, los datos se dividen en diversas tablas y, como su nombre lo indica, es imprescindible establecer relaciones entre éstas, lo cual se logra por medio de las llaves relacionales

- Llave primaria (Primary Key): Es un atributo, o conjunto de atributos que tienen que ser únicos.
- Llave secundaria (Foreing Key): Es un atributo que en alguna otra tabla es llave primaria.

4.3.3. Mapeo del diagrama entidad-relación al modelo relacional.

- Entidades: Cada entidad en un DER es transformado en una relación, los atributos de la entidad se convierten en atributos de la relación, y el atributo identificador de la entidad se convierte en la llave primaria de la relación, si se trata de una entidad débil, entre los atributos de la relación que se crea, debe incluirse la llave primaria

- Relaciones binarias: Se refieren a las indicadas por la cardinalidad, cada una tiene un método de mapeo propio.

La relación binaria de interés es uno a muchos; una vez usado el procedimiento para mapear entidades, se debe incluir la llave primaria de la entidad del lado de cardinalidad uno, como llave foránea de la relación que está del lado de la cardinalidad muchos

4.4. Esquema.

Un esquema de base de datos representa la configuración lógica de ésta, puede existir como representación visual o como un conjunto de fórmulas conocidas como restricciones de identidad, expresadas en un lenguaje de definición de datos como SQL (Inc, 2019).

Se utilizó el software MySQL Workbench para realizar el esquema como representación visual, partiendo del diagrama entidad-relación. Esta aplicación proporciona una herramienta visual que permite realizar modelado de datos. La Figura 17 muestra la zona de trabajo para la creación de una tabla.

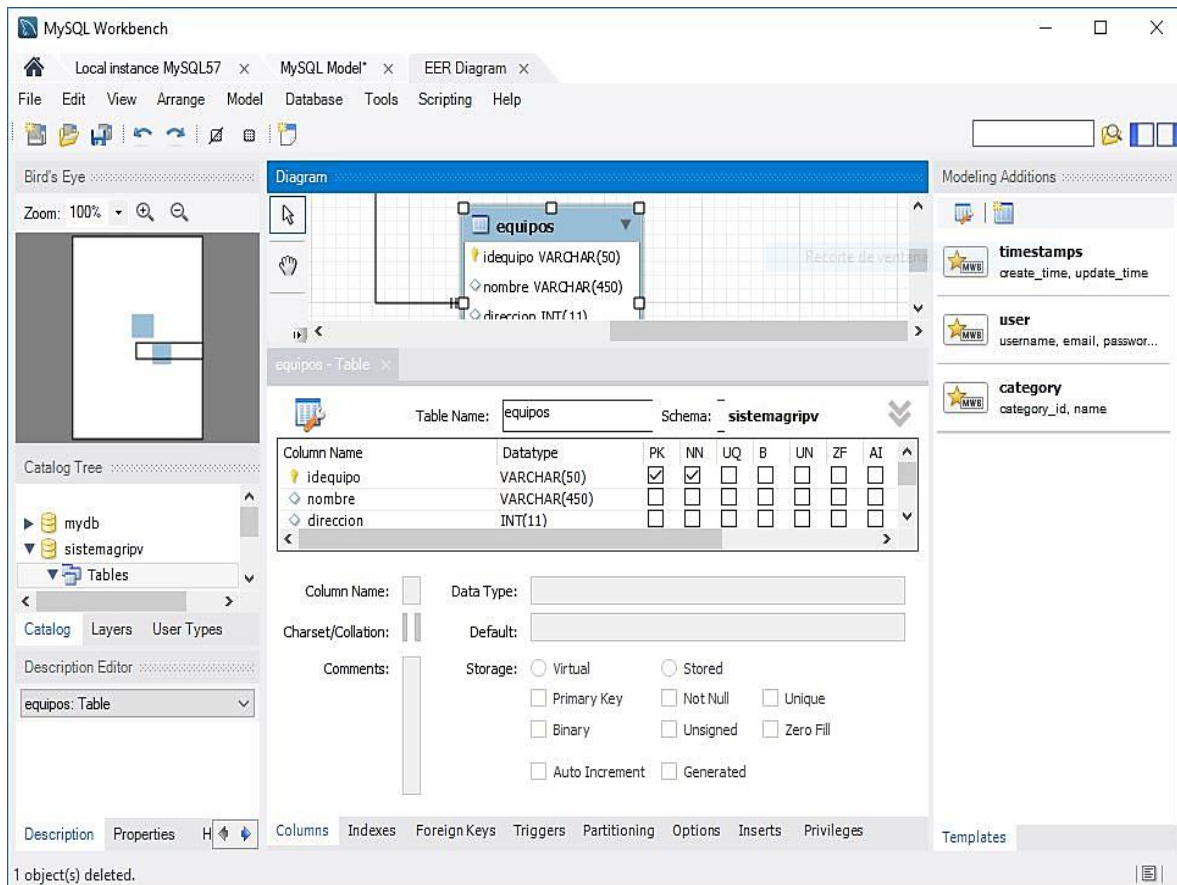


Figura 17: Zona de trabajo en MySQL Workbench

De acuerdo con el modelo relacional, la base de datos debe incluir dos tablas, una que para cada medidor especifique el nombre del equipo, el número de esclavo para la comunicación Modbus y su ubicación física, y otra que almacene los valores de las variables cada vez que se realiza una medición, indicando hora, fecha y tiempo de lectura.

Antes de realizar el esquema en MySQL Workbench, debe crearse una conexión al servidor, con el fin de poder interactuar con este.

Un servidor de bases de datos relacionales es un sistema bajo arquitectura cliente/servidor que proporciona servicios de gestión, administración y protección de la información a través de conexiones de red, gobernadas por protocolos definidos y a los que acceden los usuarios, de modo concurrente, a través de aplicaciones clientes, bien sean herramientas del propio sistema como aplicaciones de terceros (EcuRed, n.d.). El esquema realizado se presenta en la Figura 18.

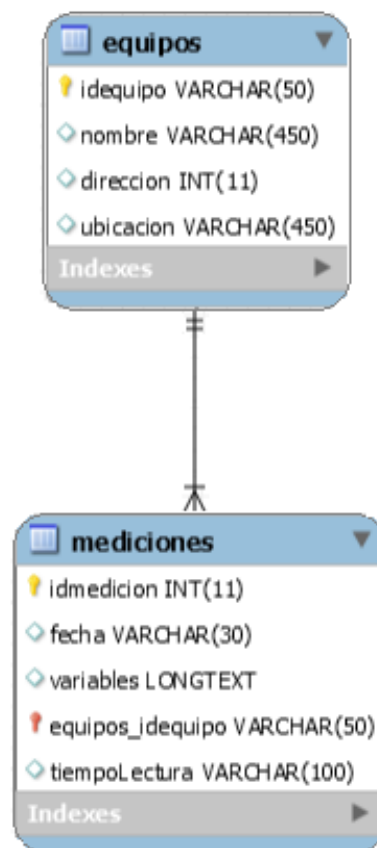


Figura 18: Esquema de la base de datos realizado en MySQL Workbench

Como puede verse, el programa identifica con un símbolo amarillo las llaves primarias y con uno rojo las llaves foráneas.

MySQL Workbench además de facilitar el diseño visual de bases de datos, permite su creación y administración, por lo tanto, la base de datos se creó a partir del esquema realizado y no fue necesario el uso del lenguaje SQL, sin embargo, una herramienta valiosa del software es la aplicación de ingeniería inversa, mediante la cual, puede obtenerse el script SQL a partir del modelo ER tal como se muestra en la Figura 19.



Figura 19: Script SQL generado a partir de la opción "Forward Engineer" de MySQL Workbench.

5. Instalación del sistema e implementación del SGBD

En este capítulo se hace un recuento del proceso mediante el cual, una vez diseñada y creada la base de datos, y desarrollado el sistema de adquisición de datos automatizado, se aprovechan las herramientas ofrecidas por los SGBD para el almacenamiento y administración de la información obtenida.

5.1. Herramientas de MySql

Los SGBD se clasifican según la forma en la que administran los datos en relacionales (SQL) o no relacionales (no SQL), teniendo en cuenta que el modelo de base de datos utilizado en este proyecto es relacional, se seleccionó el Sistema Gestor de Base de Datos MySQL para su implementación.

MySQL es ampliamente utilizado en aplicaciones creadas como software libre, entre sus ventajas está ser de fácil uso, ofrecer buen rendimiento, soporte multiplataforma y soporte SSL (Rafael Marin, 2019).

El servidor de base de datos utilizado es MySQL Server, debido a que este se accede por consola, existen herramientas que proporcionan una interfaz gráfica para facilitar esta tarea llamadas “Administradores de Bases de Datos” o DBM, por sus siglas en inglés.

Una de las opciones para bases de datos de MySQL, y la utilizada para esta implementación es SQLyog, permite, gestionar usuarios, permisos, y hacer solicitudes entre otras tareas.

El capítulo 4 describe el uso de un software de este tipo cuando se recurrió a MySQL Workbench para el diseño y creación de la base de datos.

5.2. Condiciones iniciales

Esta sección presenta los elementos que conforman el sistema de medición, las cuales constituyen el punto de partida para la realización de este trabajo de grado.

5.2.1. Cableado del bus rs485. El estándar utilizado para la comunicación Modbus es RS485, el cual define un bus para la transmisión serie multipunto, donde puede haber un equipo transmitiendo y varios recibiendo, la comunicación es semidúplex; es decir, que un equipo puede enviar y recibir, pero no al mismo tiempo.

El cableado consiste en un par de hilos sobre los que se transmite una señal diferencial para enviar los bits de datos, también pueden usarse líneas de 0 y 5 V para alimentar los dispositivos del bus.

Se admiten distancias de hasta 100 metros, lo cual aumenta las posibilidades con respecto a la ubicación de los puertos, ya sea para facilitar el acceso a estos o para conectar esclavos físicamente separados.

Esta característica se aprovecha en el sistema de medición, puesto que hay medidores instalados en la terraza y ubicar los puertos en el mismo lugar representaría una dificultad para las personas que necesitan acceder a ellos y facilitaría el deterioro de los equipos conectados a estos para la descarga de datos.

A un mismo puerto pueden conectarse 247 esclavos, la velocidad de transmisión configurada en el puerto es 19200 bps, esta debe ser igual a las de los otros dispositivos del bus (Candelas Herías, 2011).

El sistema de medición tiene tres puertos RS485, uno se encuentra en la zona de tableros del sótano del edificio y los dos restantes en el cuarto técnico del quinto piso.

5.2.2. Disposición de los puertos y medidores. Como se menciona en el Numeral 2.2, El sistema de medición de variables eléctricas está conformado por seis equipos, dos Acuvim y 4 AcuRev.

La Figura 1 presenta la ubicación inicial de los elementos del sistema de medición (equipos y puertos) en el Edificio de Ingeniería Eléctrica, y las conexiones de las líneas de comunicación.

En la zona de tableros generales de distribución ubicada en el sótano (nivel 0), se encuentran instalados dos AcuRev (Figura 20.a) y un Acuvim IIW (Figura 20.b), que tienen como propósito registrar los parámetros eléctricos necesarios para establecer y caracterizar el comportamiento energético del edificio, los 3 estaban conectados al puerto RS485 dispuesto en esta zona.

Los dos AcuRev restantes están instalados en la terraza (Figura 20.c), ambos se conectaban a uno de los puertos de comunicación ubicados en el cuarto técnico del quinto piso, estos equipos almacenan datos de energía y potencia (activa, reactiva y aparente) entregadas por los paneles fotovoltaicos, así como otras variables eléctricas (tensiones, corrientes, frecuencia, factor de potencia...). El otro puerto RS485 del quinto piso está conectado al Acuvim II ubicado en el cuarto técnico del piso 4; este equipo almacena datos correspondientes a la inyección de potencia del sistema fotovoltaico (Parrado Duque & Ordoñez Plata, 2018).



a. Equipos AcuRev instalados en la zona de tableros



b. Acuvim instalado en la zona de tableros



c. AcuRev instalado en la terraza

Figura 20: Medidores instalados

5.2.3. Descarga de datos. Para realizar la descarga de datos directamente, es necesario un computador con el software del equipo instalado, este es gratuito y está disponible en la página web del fabricante “www.accuenergy.com”. El software permite configurar las variables que almacena el equipo en memoria y los intervalos entre tomas de datos, sin embargo, una de las grandes limitantes es la baja capacidad de almacenamiento del equipo, que exige descargar datos continuamente para liberar memoria y restringe la cantidad de variables e intervalos de tiempo para guardar los valores.

Otro elemento indispensable, tanto para la descarga de datos directa como para la implementación del sistema de adquisición de datos automatizado, es un adaptador USB a RS485 como el mostrado en la Figura 21.



Figura 21: Adaptador RS485 a USB

5.3. Instalación del sistema de adquisición de datos

5.3.1. Hardware. La rutina desarrollada en el Capítulo 3 debe ejecutarse en cada uno de los puertos de comunicación, por lo tanto, usar un computador personal, como el utilizado para realizar las pruebas, implica un gasto energético alto, mayor necesidad de espacio y aumenta en gran medida los costos.

Se optó por el uso de ordenadores de placa reducida (SBC). Este tipo de dispositivos poseen la capacidad de cómputo necesaria, son de bajo costo y dimensiones reducidas. Para adquirir los dispositivos apropiados se evaluaron varias opciones, teniendo en cuenta características imprescindibles como la disponibilidad de puertos USB y comunicación por Ethernet.

La Tabla 8 relaciona los SBC disponibles en el mercado que más se ajustan a las necesidades del trabajo de grado. Debido a su precio y mayor disponibilidad en el mercado, se seleccionó la

Raspberry pi 3 (Figura 23); este modelo pertenece a la tercera generación de Raspberry pi, cuenta con CPU Quad-Core Cortex A7, posee cuatro puertos USB, un puerto Ethernet y uno HDMI.

Tabla 8.

Características de los ordenadores de placa reducida que cumplen con las especificaciones.

DISPOSITIVO	PROCESADOR	RAM	ETHERNET	USB	COSTO (COP)
Odroid-c2	ARM A53 4x2 Ghz	2G	si	si	370.000
Beaglebone Black	ARM A8 1 Ghz	1G	si	si	280.000
Hummingboard	ARM A9	2G	si	si	260.000
Orange Pi Plus	ARM A7 4x1.2 Ghz	1G	si	si	240.000
Raspberry pi 3	ARM V8	1G	si	si	175.000



a. Raspberry pi 3



b. Raspberry pi 3 con carcasa y ventilador

Figura 22: Tarjetas adquiridas para la implementación

La tarjeta Raspberry no incluye memoria interna, cargador para alimentación ni carcasa, por lo tanto, también fue necesario adquirir estos elementos.

Inicialmente, el servidor fue instalado en un ordenador portátil que, al igual que las tarjetas Raspberry, se conecta a la red LAN institucional, su ubicación puede variar sin afectar el funcionamiento si se mantiene la conexión, ya sea WIFI o por Ethernet.

A pesar de estar conectados a la misma red, el servidor de base de datos es remoto para todos los equipos, excepto en aquel donde se encuentra instalado. Las conexiones entre los dispositivos que conforman el hardware del sistema de adquisición de datos, y su ubicación, se representan en la Figura 23.

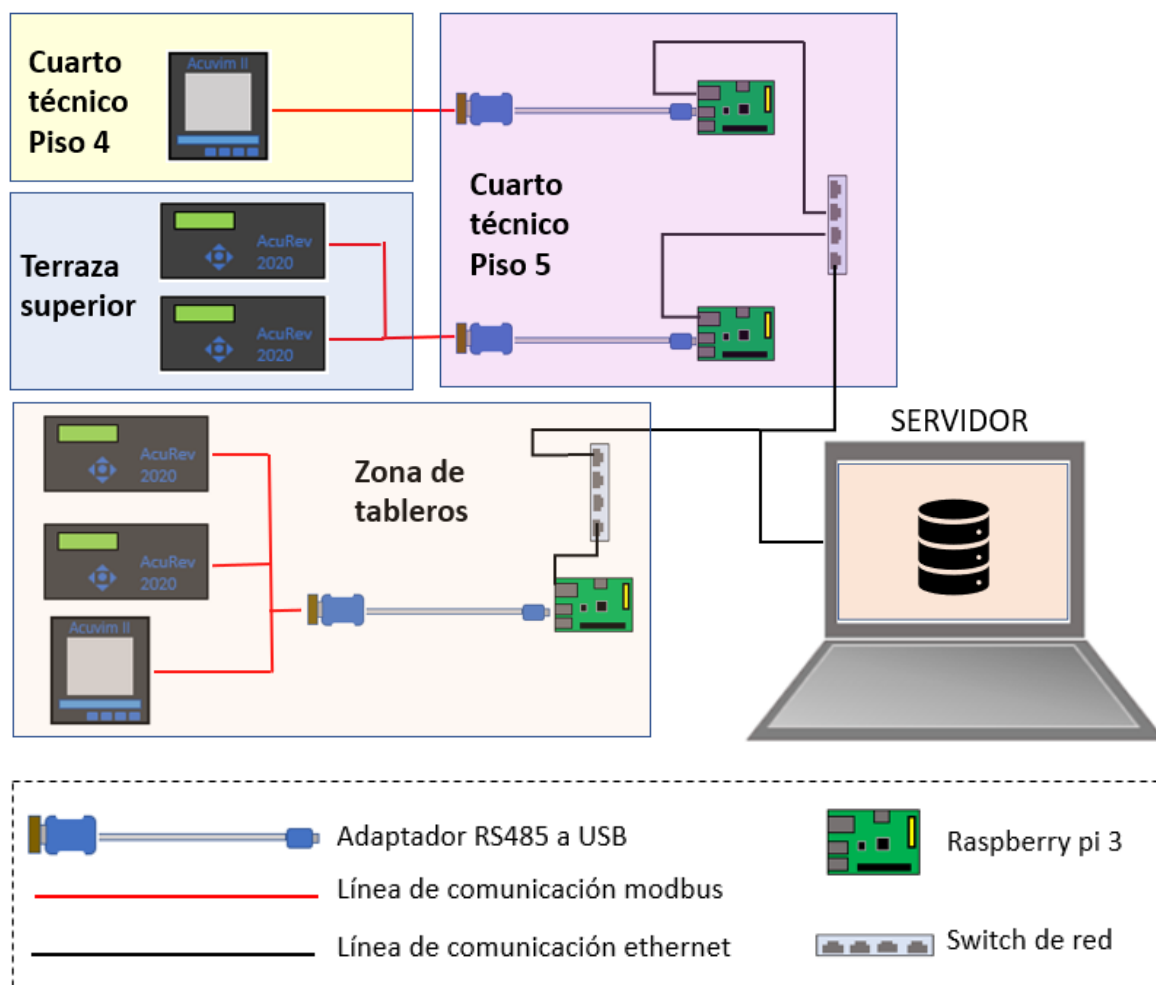


Figura 23: Hardware inicial del sistema de adquisición de datos.

5.3.2. Software. La tarjeta Raspberry Pi soporta diferentes sistemas operativos, se optó por la instalación de Raspbian, un sistema operativo libre basado en la distribución de Linux llamada Debian, que ha sido optimizado para el hardware de Raspberry Pi.

La implementación exige realizar dos cambios importantes en la rutina desarrollada previamente, el primero corresponde a la cantidad de equipos de los cuales se realizan lecturas en cada solicitud, ya que en dos de los puertos debe conectarse más de un medidor, por lo tanto, es necesario crear un objeto de lectura para cada medidor.

Para facilitar la identificación de los equipos, se asignó una dirección de esclavo diferente a cada uno, a pesar de que esto sólo es una exigencia cuando están conectados al mismo maestro.

Se configuró la rutina para que permita seleccionar entre el total de los esclavos aquellos con los cuales se va a establecer comunicación, se incluyó entre los datos de entrada el nombre de los equipos en la base de datos, tal como en la primera versión, debe ingresarse el tipo de equipo para cada uno de los esclavos.

La interfaz de usuario de esta versión se muestra en la Figura 24

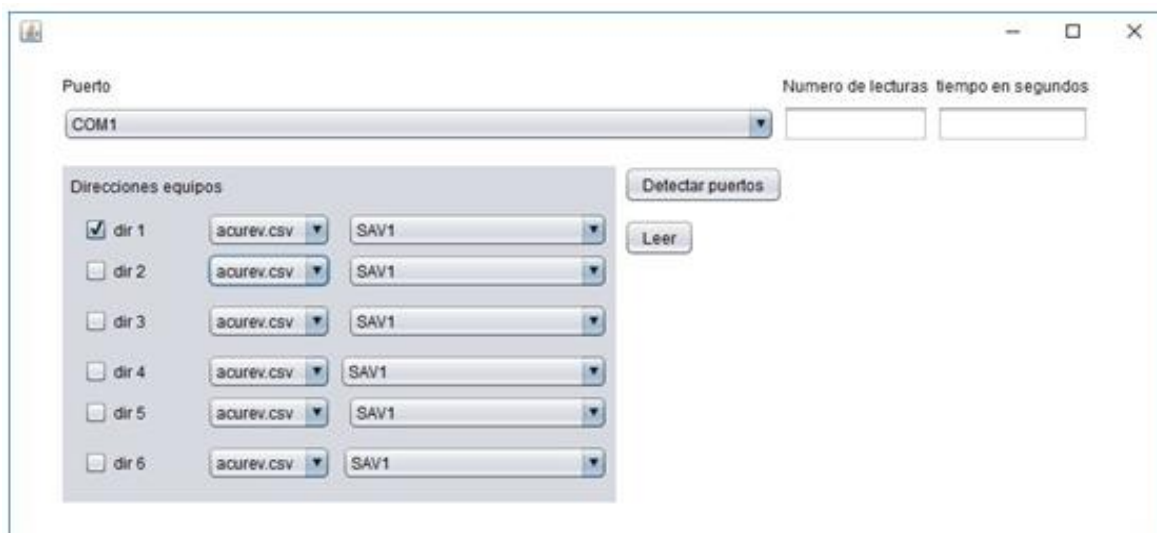


Figura 23: Interfaz de usuario de la rutina para adquisición de datos

Al ejecutar la nueva rutina, se realiza en una solicitud, tantas lecturas como equipos seleccionados, estas se ejecutan en orden ascendente según el número de esclavo, y una vez concluida la comunicación con un equipo, se inicia con el siguiente.

El segundo cambio realizado para obtener la versión final de la rutina consistió en la conexión con la base de datos, para esto se usa un túnel SSH, que se autentica con las credenciales del servidor (usuario y contraseña).

Para que los programas JAVA accedan a Sistemas de gestión de bases de datos, se utiliza una Interfaz de programación de aplicaciones (API) llamada JDBC, que permite establecer conexión con la base de datos, enviar sentencias SQL y procesar los resultados (IBM, 2013).

Para configurar el servidor como remoto, se ingresan las credenciales (usuario y contraseña) que debe solicitar para permitir ingreso de información desde dispositivos diferentes al cual se instaló el servidor de base de datos.

5.4. Pruebas del sistema implementado

La primera prueba consistió configurar el servidor de forma local; para lo cual, fue instalado en el mismo equipo que ejecutaba la rutina para adquisición de datos. Esto con el fin de verificar el buen funcionamiento de la conexión a base de datos agregada a la rutina. Se realizó una comparación entre los datos almacenados y los descargados mediante el software y los resultados fueron favorables.

Una vez incorporadas las tarjetas al sistema de medición; éste fue puesto en marcha tomando datos cada 10 minutos durante 3 días, el almacenamiento en la base de datos no presentó inconvenientes; sin embargo, presentó un desfase en tiempo, este error también surgió en las

pruebas de la primera versión de la rutina, y fue corregido restando el tiempo de lectura al intervalo entre éstas entregado por el usuario.

En este caso, el desfase surgió por el tiempo que tarda la escritura en base de datos, y fue corregido de la misma forma. Una vez realizado este cambio, el sistema fue puesto en marcha nuevamente, la Figura 25 presenta parte de la base de datos durante la prueba, en la columna “fecha” puede verse la hora de inicio de cada lectura.

1 Messages

2 Table Data

Figura 24: Mediciones almacenadas en la base de datos sin desfase de tiempo

Es importante tener en cuenta que la hora y el tiempo de lectura; corresponden a las tarjetas y no a los medidores; por lo tanto, en los puertos que se comunican con más de un esclavo, las lecturas tardan entre 30 y 36 segundos.

Durante la prueba se presentó una falla en la red eléctrica del edificio y el sistema operativo de las tarjetas se reinició; la información había sido almacenada correctamente hasta antes de este evento; sin embargo, la toma de datos se interrumpió.

Para las tarjetas ubicadas en el piso 5 (Raspberry 1 y Raspberry 2), el inconveniente se resuelve conectándolas al Sistema de Alimentación Ininterrumpida (UPS) dispuesto en este espacio.

5.5. Mejoramiento del sistema y versión final

Las pruebas realizadas permitieron identificar principalmente las siguientes falencias :

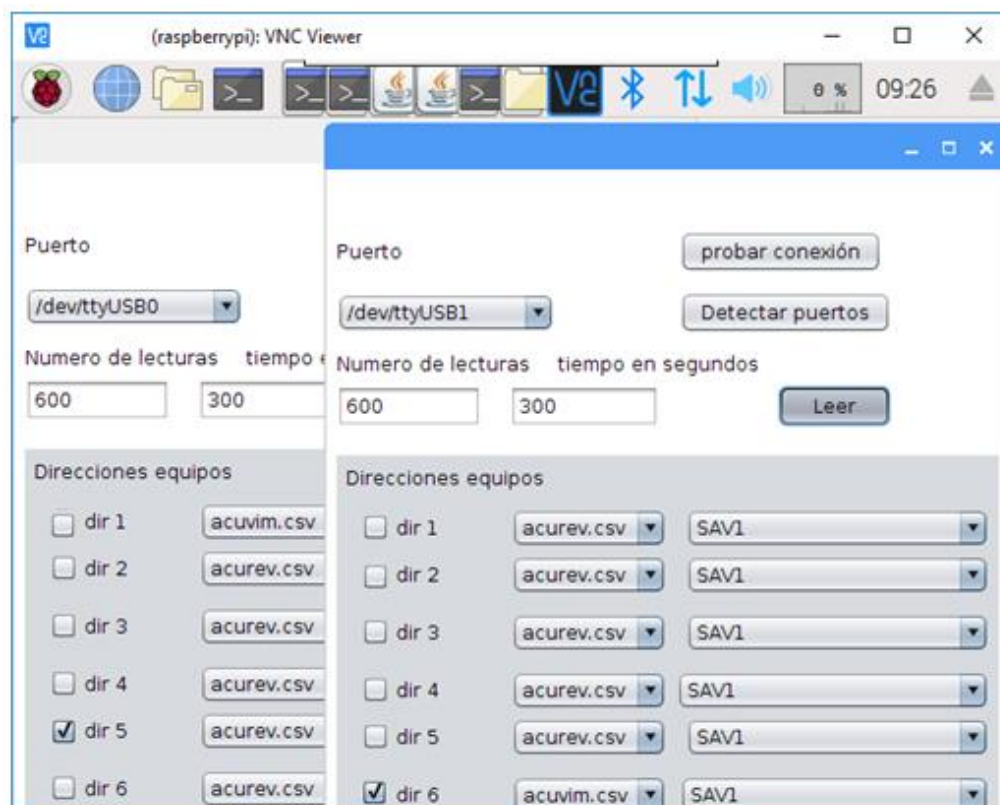
- La tarjeta número tres no está conectada a un sistema de alimentación ininterrumpida, por lo tanto es vulnerable a eventos que causen el reinicio de su sistema operativo
- Debido a las características del protocolo de comunicación, no es posible solicitar todas las variables de un equipo simultáneamente. Esto no representa mayor problema cuando se hacen solicitudes a un solo equipo, ya que el tiempo de lectura por cada variable es aproximadamente 100 ms, sin embargo, cuando se comunican varios esclavos mediante una sola tarjeta, la lectura puede llegar a tardar 36 segundos.

Para solucionar el primer ítem, se propuso desplazar la tarjeta Raspberry 3 al cuarto técnico del piso 5 para alimentarla mediante la UPS, esto implica trasladar los puertos modbus, la propuesta es viable puesto que modbus admite hasta 100 metros de cableado entre el esclavo y el puerto.

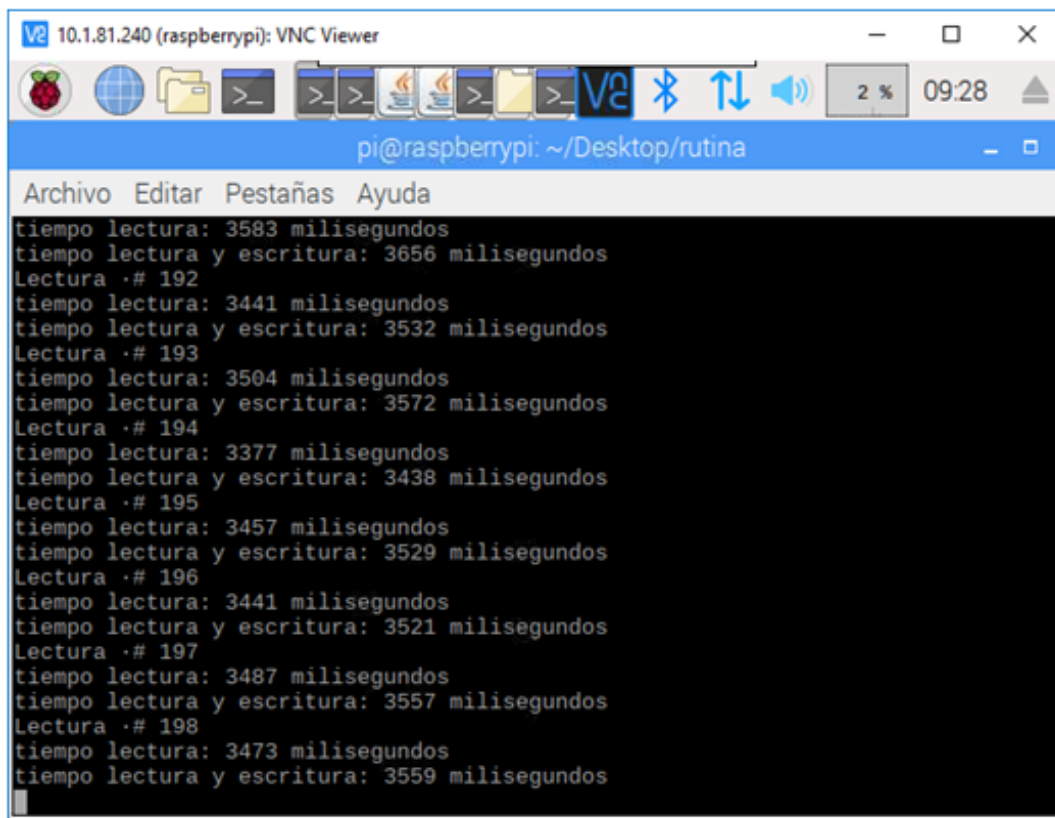
La disposición de las tres tarjetas en el mismo espacio abre la posibilidad de independizar los puertos modbus con el fin de correr dos rutinas simultáneamente en la misma tarjeta, solucionando el problema de sincronización.

Antes de realizar el cambio, se comprobó la capacidad de la tarjeta Raspberry para correr dos rutinas en paralelo utilizando los esclavos número 5 (AcuRev) y número 6 (Acuvim); que ya estaban conectados a distintos puertos. Se realizaron 600 lecturas con intervalos de 5 minutos como se muestra en la Figura 26.a. La prueba culminó sin inconvenientes.

Como se aprecia en la Figura 26.b, los tiempos de lectura del Acuvim oscilan en el mismo rango que en pruebas anteriores, cuando éste era el único esclavo conectado a la tarjeta.



a: Interfaz de la tarjeta Raspberry para lanzar dos rutinas simultáneamente.



```
tiempo lectura: 3583 milisegundos
tiempo lectura y escritura: 3656 milisegundos
Lectura .# 192
tiempo lectura: 3441 milisegundos
tiempo lectura y escritura: 3532 milisegundos
Lectura .# 193
tiempo lectura: 3504 milisegundos
tiempo lectura y escritura: 3572 milisegundos
Lectura .# 194
tiempo lectura: 3377 milisegundos
tiempo lectura y escritura: 3438 milisegundos
Lectura .# 195
tiempo lectura: 3457 milisegundos
tiempo lectura y escritura: 3529 milisegundos
Lectura .# 196
tiempo lectura: 3441 milisegundos
tiempo lectura y escritura: 3521 milisegundos
Lectura .# 197
tiempo lectura: 3487 milisegundos
tiempo lectura y escritura: 3557 milisegundos
Lectura .# 198
tiempo lectura: 3473 milisegundos
tiempo lectura y escritura: 3559 milisegundos
```

b: Tiempos de lectura para el esclavo número 6 durante el desarrollo de la prueba

Figura 25: Prueba de funcionamiento de dos rutinas corriendo simultáneamente en la misma tarjeta.

La Figura 27 presenta un esquema de la disposición final de los elementos que conforman el hardware del sistema de adquisición de datos.

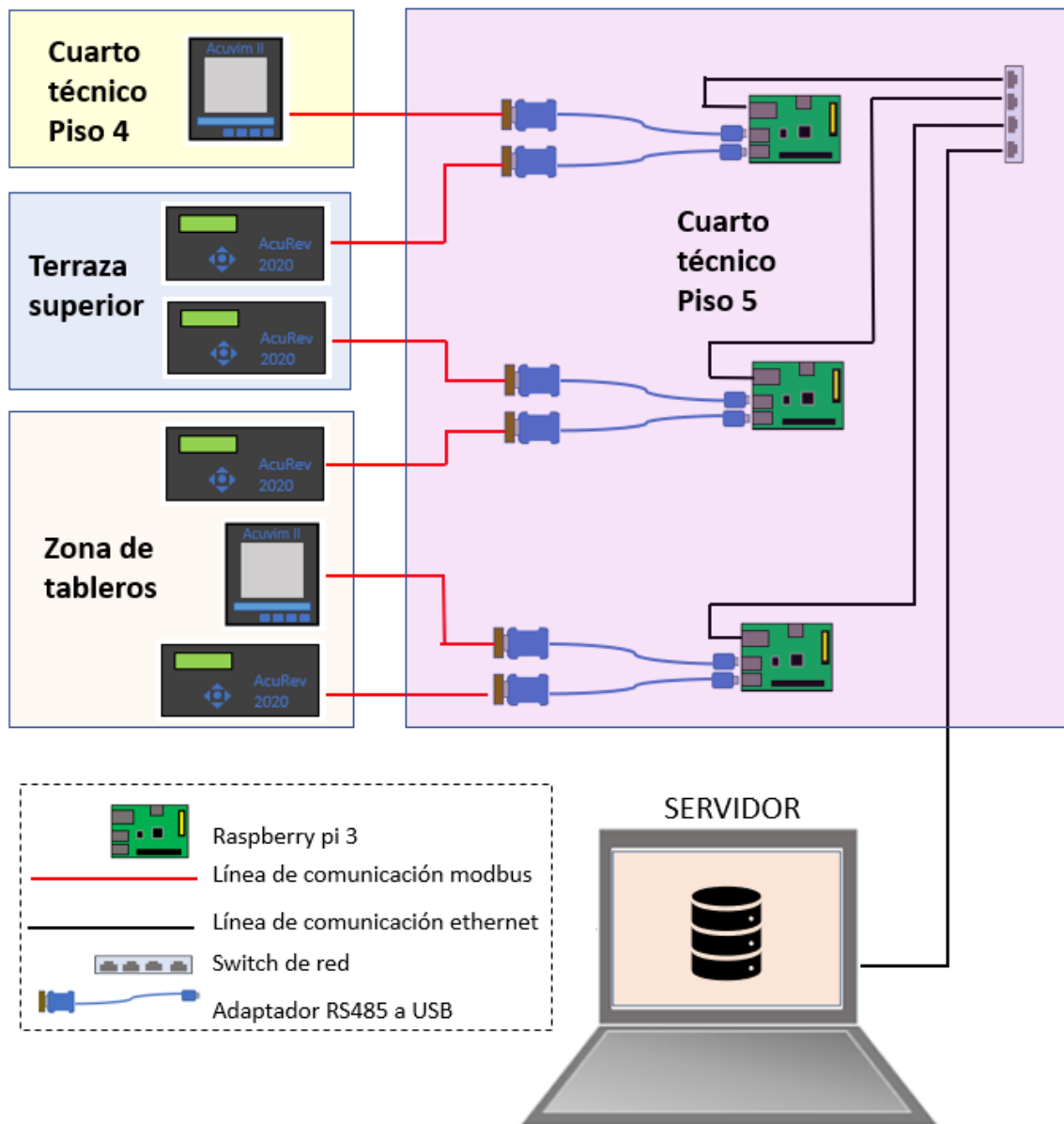


Figura 26: Disposición final de los elementos del sistema de adquisición de datos.

5.6. Equipos y costos

La Tabla 9 relaciona los costos de los dispositivos adquiridos especialmente para la realización de este trabajo de grado; no se incluyen los precios de los medidores ni el computador en el que se aloja el servidor físico puesto que habían sido previamente adquiridos.

Tabla 9.

Equipos y costos.

ELEMENTO	CANTIDAD	COSTO UNITARIO	TOTAL
Raspberry Pi 3	3	175.000	525.000
Adaptador Raspberry 5V 3A	3	27.000	81.000
Memoria micro SD 16Gb clase 10	3	20.000	60.000
Switch internet	2	35.000	70.000
Adaptador R485 a USB	6	166.000	996.000
COSTO TOTAL:			1.732.000

6. Conclusiones

La información recolectada por el sistema desarrollado es altamente confiable; las diferencias entre las mediciones realizadas mediante Modbus y las descargadas mediante el software del equipo se deben especialmente a que las primeras son medidas instantáneas y las segundas son un promedio, y a que los datos extraídos por el sistema desarrollado incluyen más cifras significativas.

Para aquellas variables que no presentan cambios instantáneos relevantes el error es inferior al 5%.

El software es flexible ante la cantidad de equipos que pueden incorporarse, una vez resueltos los detalles del hardware, para agregar un equipo debe incluirse una nueva tupla en la tabla “EQUIPOS” de la base de datos y, en caso de que este sea de diferente tipo (Acuvim o Acurrev), preparar el documento de entrada.

El protocolo de comunicación utilizado no permite leer todas las variables de un equipo en el mismo instante de tiempo, pero la duración de la lectura es aproximadamente 3 segundos en el caso del Acuvim, y de 15 en el caso del Acurrev. Si el intervalo entre mediciones continúa siendo 10 minutos, la diferencia entre el tiempo de lectura de la primera variable y la última corresponden al 0.5% y al 2.5% del intervalo entre medidas respectivamente, lo cual no representa un error significativo.

Los equipos almacenan datos en memoria con respecto a la hora guardada en uno de sus registros; los desfases de tiempo son un problema común que además puede identificarse de forma tardía, cuando se va a realizar la descarga, la lectura de datos mediante Modbus es

independiente de la hora almacenada en los equipos; por lo tanto, el sistema no es susceptible a este desfase.

Se logra la sincronización de los datos y su almacenamiento en una base de datos centralizada, facilitando el acceso a la información y aumentando su seguridad, tal como se planteaba en los objetivos de este trabajo de grado

7. Recomendaciones

La capacidad y velocidad de almacenamiento pueden mejorar ampliamente si la base de datos es migrada a un servidor con mejores características.

Sin un sistema de alarmas, los errores de funcionamiento que pueden producirse por fallas en la red Ethernet o por eventos relacionados con el hardware, son detectados de forma tardía. Se recomienda implementar alarmas para alertar a los administradores de la base de datos cuando se presenten comportamientos inesperados, con el fin de identificar el problema y corregirlo oportunamente, minimizando la pérdida de información.

Referencias Bibliográficas

- Arias, Á. (2014). *Bases de Datos con MySQL* (2nd ed.). IT Campus Academy.
- Candelas Herías, F. A. (2011). Comunicación con RS-485 y MODBUS, (37800), 1–28. Retrieved from <https://rua.ua.es/dspace/bitstream/10045/18990/1/AA-p3.pdf>
- Corp, A. (2014). www.accuenergy.com. Retrieved from <https://www.accuenergy.com/files/acuvim-ii/Acuvim-II-1040S1101-Spanish.pdf>
- Cortazar Bellas, F., Martinez Ananue, R., & Fresno Fernandez, V. (n.d.). *Lenguajes de Programación y Procesadores*.
- EcuRed. (n.d.). Servidores de Base de Datos. Retrieved April 15, 2019, from https://www.ecured.cu/Servidores_de_Base_de_Datos#Servidores_de_Bases_de_Datos_MySQL
- Garrido Avaenza, P. P. (2015). *Comenzando a programar con JAVA*.
- Hui, S. ., & Chan, S. C. (2011). Integration of green roof and solar photovoltaic systems. In *Joint Symposium* , (November 2011), 1–12. Retrieved from <https://www.researchgate.net/publication/281901499>
- IBM. (2013). IBM Knowledge Center. Retrieved from https://www.ibm.com/support/knowledgecenter/es/SSGU8G_12.1.0/com.ibm.jdbc_pg.doc/ids_jdbc_011.htm
- Inc, L. S. (2019). Lucidchart. Retrieved April 11, 2019, from www.lucidchart.com/pages/es/que-es-un-esquema-de-base-de-datos

Monge Gómez, J. M. (n.d.). *Diseño e implementación de una interfaz MODBUS para la integración de grupos electrógenos a la plataforma de gestión SCADA.*

Parrado Duque, A., & Ordoñez Plata, G. (2018). Propuesta de arquitectura de medición inteligente en sistemas de distribución con generación distribuida en redes eléctricas de baja tensión. *Universidad Industrial de Santander.*